LIBRARY OF THE
UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN

510.84
Il6r
no.559-564
cop.2

[redacted]

Digitized by the Internet Archive
in 2013

http://archive.org/details/algorithmmodels562hoff

Il6r
no. 562
cop. 2

UIUCDCS-R-73-562

Math

ALGORITHM MODELS

by

DONALD JAY HOFFMAN

MAY 14 1973

February 1973

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

THE LIBRARY OF THE
MAY 2 1973
UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN

UIUCDCS-R-73-562

ALGORITHM MODELS*

by

DONALD JAY HOFFMAN

February 1973

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
URBANA, ILLINOIS 61801

* This work was supported in part by the Joint Services Electrical Program (US Army, US Navy, US Air Force) under contracts DAAB-07-67-C-0199 and DAAB-07-72-C-0259; advised by Prof. S. R. Ray, Department of Computer Science and submitted in partial fulfillment of the requirements of the Graduate College for the degree of Master of Science in Computer Science.

ACKNOWLEDGMENT

The author would like to express his great appreciation to Dr. Sylyian Ray for his help and guidance in conceiving and working out this problem. My appreciation also goes to Mrs. Barbara Bunting for typing the manuscript.

TABLE OF CONTENTS

Page

1. INTRODUCTION 1

2. MODEL GENERATING AND IMPLEMENTATION CHECKING PHILOSOPHY . . 2

    2.1 Approach 2
    2.2 Method. 2
    2.3 Implementation Analysis 3
    2.4 The Algorithm Checker (AC) 3

3. EXAMPLE - METHOD OF FALSE POSITION 4

    3.1 Description of the Algorithm 4
    3.2 Create the Model 4

4. THE ALGORITHM CHECKER 10

    4.1 General Physical Description. 10
    4.2 Implementation Method of False Positions 10

5. CONCLUSION 16

APPENDICES

1. Implementation Restrictions - method of false position. 17

2. The Algorithm Checker Processing Some Test Implementations of the Method of False Position 19

3. The Algorithm Checker Program 32

1. INTRODUCTION

A problem has always existed in basic computer science education in training students to take a problem and translate it into a working solution on a computer. This process can be broken down into three steps:

1. Create an algorithm
2. Make a flow chart of the algorithm
3. Code the algorithm

A major area of intellectual difficulty is the process of conceptualizing an algorithm in a form that can be easily translated into a flow chart. There has not been and perhaps never will be a rigorously prescribed way to approach this problem.

The human mind performs these processes in a way yet unknown. What I will try to do is treat the human mind as a black box and perform the same operations by creating a system of logical cognitive models of the required operations in taking an algorithm and translating it into a form suitable for coding. I will then demonstrate these cognitive models by demonstrating a method of checking the logical construction of implementations of some algorithms by using these models as part of a system for automatic fault detection. My specific approach will be:

1. Develop a generalized heuristic procedure for generating conceptual models of algorithms;
2. Develop these conceptual models for some common algorithms.
3. Demonstrate step 1 and step 2 by using these models in a preprocessor written to perform automatic fault detection on implementations of these algorithms.

2. MODEL GENERATING AND IMPLEMENTATION CHECKING PHILOSOPHY

Decide which features are important and devise a technique for algorithm detection.

2.1 Approach

Separate the algorithm into a series of partitions that are each a logical operation and are easily physically separated.

If the partitions are large and/or very complex then perform the separation operation again, and so on.

2.2 Method

Represent the algorithm as a Type 3 language represented by a series of productions. Describe the language using several levels of non-terminals - the lower the numerical level assigned to the non-terminal the more complex is the sublanguage the non-terminal represents.

2.3 Implementation Analysis

Define a function, plausibility, whose range is between 0 and 100, and whose initial value is 100. The plausibility will be an estimate of the likelihood that a specific implementation of an algorithm is an accurate mechanization of the algorithm in question. The plausibility is determined by two factors:

1. The existence of certain features of the algorithm in the implementation in question.
2. The physical location, precedence, of the various features in this implementation.

The existence and precedence relations can both be discovered by analyzing the productions in the model. Each feature is assigned a step constant whose range is between 0.0 and 1.0. It is an estimate (heuristic) of the relative importance of this feature in the algorithm model. The absence of this feature or its physical misplacement in the implementation shall suggest an error in the implementation.

When checking the implementation, should such an error condition arise, the plausibility is reduced by multiplying it by the associated step constant.

2.4 The Algorithm Checker (AC)

The Algorithm Checker is a program written for the IBM 360/75 computer in SNOBOL language. It implements the above techniques to perform analysis of the implementations of some specific algorithms. The input to AC is a student program, coded as a FORTRAN subroutine intended to mechanize an algorithm which AC has been written to check. If AC is told which algorithm to check for AC will be able to:

1. Estimate the plausibility that the algorithm is being correctly implemented or not.
2. Recognize and comment on logical errors in the input program.

3. EXAMPLE - METHOD OF FALSE POSITION

3.1 Description of the Algorithm

Method of False Position

One approach to the solution of non-linear equations will be to use linear inverse interpolation to derive methods of the form:

$$x_{i+1} = F(x_i\ x_{i-1}, \ldots, x_{i-n+1}).$$

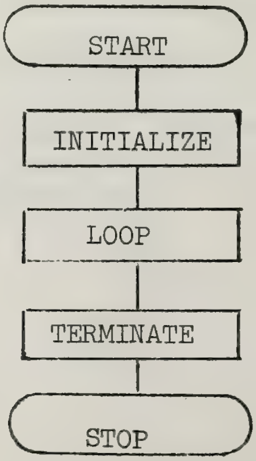
One such method is the method of false position. Suppose we can find two points x_A and x_B such that $f(x_A)*f(x_B)<0$. The chord joining y_A and y_B intersects the x axis at a point x_N. Then chosing x_N and x_i (either x_A or x_B) such that $f(x_N)*f(x_i)$ 0. We repeat the same procedure to obtain a new x_N and so on. Terminate this process when f_N is sufficiently small.

3.2 Create the Model

A. The method of false position algorithm can be broken down into steps:

1. Initialization
2. Loop
3. Terminate

These steps should occur in the above order both logically and physically with a high probability and the algorithm can be represented by the following flow chart:



Or it can be described by this production:

< false position> ::= < initialize> < loop> < terminate>

Assign level zero to the non-terminal on the left hand side and assign level one to the three non-terminals on the right hand side.

From this production we can immediately derive a precedence relationship where the three steps should appear in this specific order with a very high liklihood.

Precedence Table

A \ B	Initialize	Loop	Terminate
Initialize	0	-1	-1
Loop	1	0	-1
Terminate	1	1	0

-1 A must precede B

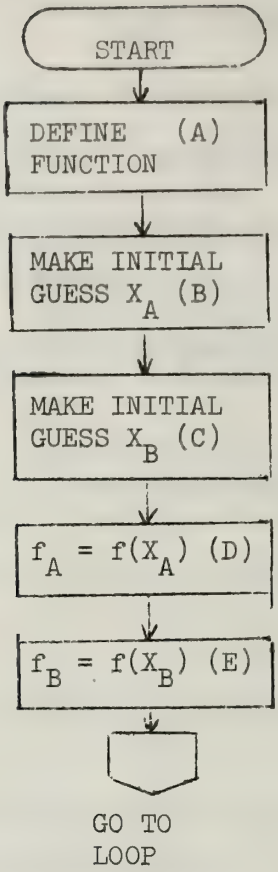
0 don't care

+1 B must precede A

B. The initialization step should perform these operations:

1. The definition of the non-linear function, F. If this operation is present, in FORTRAN, it would most likely be a function statement.
2. Make first two guesses at the root of F, X_A and X_B.
3. Calculate the values of F at X_A and X_B, f_A and f_B.

The initialize step could be represented by the following flow chart:



However the possible orders of operation for the various operations in the initialization step are not unique; there could be several different flow charts for the initialization step. The initialization step can be represented by just one set of productions:

< initialize> ::= < define function> < initialize X_A>

(< initialize X_B> (< calculate f_A> < calculate f_B>) |

(< calculate f_B> < calculate f_A>)) |

(< calculate f_A> < initialize X_B> < calculate f_B>)

This production is unique. The left-hand side of the production is level one, from before, and the non-terminals on the right-hand side are assigned level two. From this production we can derive the following precedence relations:

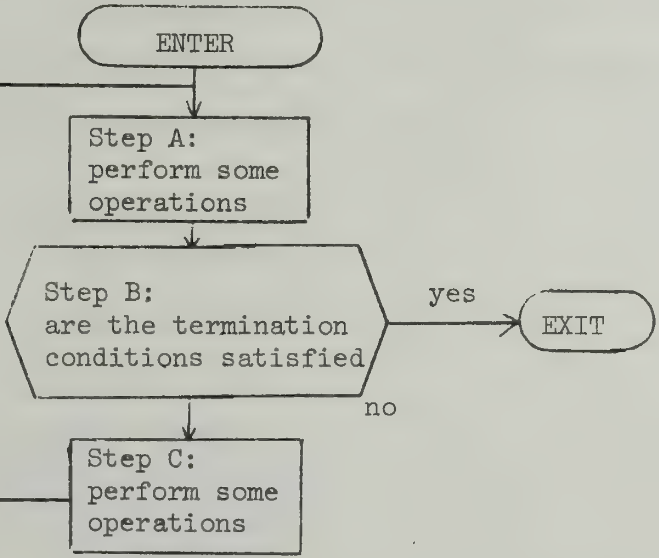
PRECEDENCE TABLE

Y X	A	B	C	D	E
A	0	-1	-1	-1	-1
B	1*	0	0	-1	0
C	1*	0	0	0	-1
D	1	1	0	0	0
E	1	0	1	0	0

+1	Y must precede X
0	don't care
-1	X must precede Y

* note that step A precedes steps B and C strictly due to FORTRAN syntax requirements.

C. An iterative process can be described by a generalized loop step. This generalized loop step can be represented by the following flow chart.

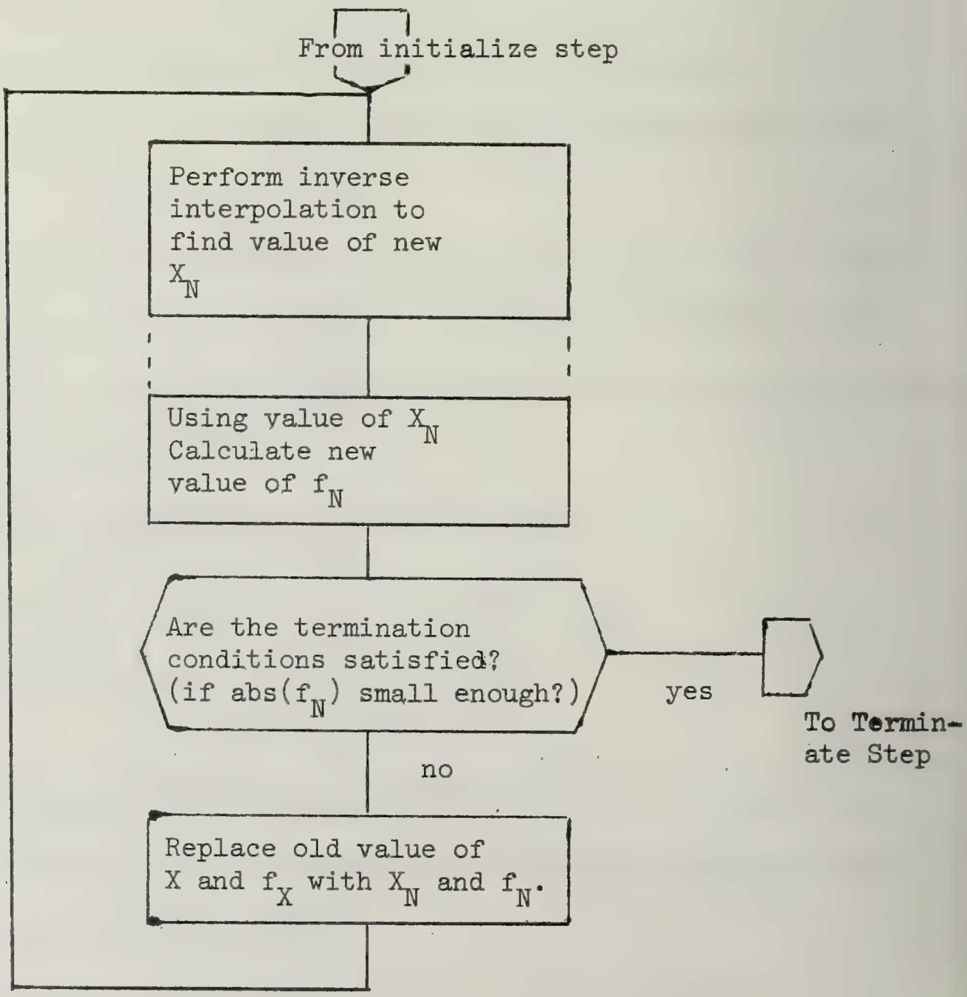


Note that either step A or step C may be null. However for the method of false position both steps should exist.

The loop step should perform these operations:

1. Calculate the next guess at the root of F, X_N.
2. Calculate the value of the function F, at X_N, f_N.
3. Check the value of f_N, (or some other function of X_N) if it is small enough, then exit the loop.
4. Decide which old value of X to replace for use in the calculations in the next pass through the loop. Then replace either X_A or X_B and the associated f_A or f_B with X_N and f_N respectively.

Substituting into the generalized loop step above, the loop step of the method of false position can be represented by the following flow chart:



From initialize step
Perform inverse interpolation to find value of new X_N
Using value of X_N Calculate new value of f_N
Are the termination conditions satisfied? (if abs(f_N) small enough?)
yes
To Terminate Step
no
Replace old value of X and f_X with X_N and f_N.

Each operation in this flow chart can be expected to be found in the logical position corresponding to its position in the flow chart with a high probability. The physical position os these operations should also be similar to the flow chart with a somewhat lower probability. The operations that must be performed to satisfy the requirements of the box are physically unique; there is more than one way to perform any operation.

The loop step of the method of false position algorithm can be represented by the following production:

< loop> ::= < calculate new X_N, calculate f_N from X_N>

< branch if f_N small enough> < test signed f_N, replace X and f>

From the above production we can derive a precedence relationship for all elements of the loop step. In this case the flow chart above is the only probable way to order the steps.

D. The termination step of the algorithm is not well defined. The only requirements on it are exclusively syntactic. The termination step must be ended by a RETURN statement followed by an END statement.

4. THE ALGORITHM CHECKER

4.1 General Physical Description

The Algorithm Checker program will be coded in IBM/360 SNOBOL IV. It consists of a supervisor program which calls various subprograms written to check for specific features.

AC will search for the specific features of an algorithm in a sequence of operations derived from the productions in the algorithm model. AC will first separate the major program sections (level 1 non-terminals) and then search these major program sections for subsections specified by level 2 non-terminals, and so on. While this is being performed AC will also be continuously updating the plausibility of this implementation each time an error condition occurs.

4.2 Implementation Method of False Positions

A. Level 1

The first thing AC must do is separate the input program into the three distinct steps specified by the production:

< false position> ::= < initialize> < loop> < terminate>

This step breakdown operation can most easily be performed by first looking for distinctive physical code structures in the various steps. The most distinctive and easiest-to-detect feature is the structure of an iterative operation. In the method of false position there is only one iterative operation expected, the loop step.

The loop step is found by scanning the input program for one of these patterns:

Pattern 1 *

```
n ........
  ........
  ........
IF(A < relational operator>  B)GOTOn
```

Pattern 2

```
n ........
  ........
  ........
IF(A < relational operator> B)GOTOm
  ........
  ........
  GOTOn
m ........
```

There are other more complex ways of coding iterative operations but for now I will assume that only one of these more simple patterns has been used.

Should a pattern match be successfully performed, it will be assumed that all code physically preceding the loop step is the initialize step and that all code physically following the loop step is the terminate step.

If the loop step cannot be successfully isolated it becomes extremely difficult for AC to proceed beyond this point and obtain any meaningful results. So, if the loop step cannot be detected and isolated, AC will reduce its plausibility estimate to a very low value and will terminate processing this implementation.

---

* < relational operator> ::= .LT.,.LE.,.GT.,.GE

B. Level 2 - Initialize

The initialize step was defined by the productions:

$$< \text{initialize} > ::= < \text{define function} > < \text{initialize } X_A >$$

$$(< \text{initialize } X_B > (< \text{calculate } f_A > < \text{calculate } f_B >) \mid$$

$$(< \text{calculate } f_B > < \text{calculate } f_A >)) \mid$$

$$(< \text{calculate } f_A > < \text{initialize } X_B > < \text{calculate } f_B >)$$

The left-hand side of the production is a level 2 non-terminal and the right-hand side is composed of level 3 non-terminals. The level 3 non-terminals, in this case, will all reduce to terminals. The terminals for this language are all equivalent to one or more FORTRAN statements.

$$< \text{define function} > ::= \{ var_1(var_2) = exp(var_2) \}$$ is

one such level 3 production. var_1 is the name of the function of var_2 is the name of the functions dummy variable. Similarly

$$< \text{initialize } X_A > ::= \{var_3 = exp\} \mid \{var_3 = const\}$$

$$< \text{initialize } X_B > ::= \{var_4 = exp\} \mid \{var_4 = const\}$$

$$< \text{calculate } f_A > ::= \{var_5 = var_1(var_3)\}$$

$$< \text{calculate } f_B > ::= \{var_6 = var_1(var_4)\}$$

Where var_3 is X_A, var_4 is X_B, var_5 is f_A, var_6 is X_B, exp is any valid FORTRAN arithmetic expression, $exp(var_n)$ is any valid FORTRAN arithmetic expression which contains var_n, and const is any valid FORTRAN constant. All complete FORTRAN statements are enclosed in braces.

We scan the code searching for the patterns specified by the above productions. If we have pattern matches we assign the names of the variables in the code to the variables X_A, X_B, f_A, f_B, respectively. If any of the pattern matches fail, or the operations are out of correct order, the plausibility is reduced by multiplying it by the appropriate step constant.

If DELTA, the loop termination control value (see next section), is initialized in the initialize step. There are some additional complications caused by certain ambiguities. We must diverge from a completely systematic approach and use some additional heuristics and first determine the additional value of DELTA.

C. Level 2 - Loop

The logical expression in the IF-GOTO, {IF (logical expression) GOTOn}, statement which controls loop termination should reduce to the form: (A.LT.DELTA) where A is a positive valued expression and DELTA is a positive constant such that when the expression is true the loop step is exited. However, there are several equivalent forms of this expression which are not immediately identifiable. For the expression A.GT.B. assuming that A and B are variables of date type REAL. Here is a list of some equivalent expressions:

```
A.GT.B
A.GE.B
B.LT.A
B.LE.A
.NOT.(B.GT.A)
.NOT.(B.GE.A)
.NOT.(A.LT.B)
.NOT.(A.LE.B)
A-B.GT.0
A-B.GE.0
B-A.LT.0
B-A.LE.0
```

I will consider the relational operations .LT. equivalent to .LE., and .GT. equivalent to .GE. because for real numbers of finite length due to computer truncation error, the relational operators .LT. and .LE., and .GT. and .GE., are indistinguishable.

The loop step can be represented by the following production:

< loop> ::= < calculate new X_N, calculate f_N from X_N>

< branch on condition satisfied> < replace old X and associated f>

The level 3 non-terminals on the right-hand side of the above reduce to:

< calculate new X_N, calculate f_N from X_N> ::= < calculate X_N>

< calculate f_N>

$$< \text{calculate } X_N> ::= \{ var_7 = \frac{x_A f_A - X_B f_B}{f_A - f_B} \}$$

where var_7 is X_N and X_A, X_B, f_A, and f_B were defined in the initialization step.

< branch on condition satisfied> ::= { IF (logical expression) GOTOn}

< calculate f_N> ::= { $f_N = f(X_N)$}

< replace old X and associated f> ::= < test> < replace a>

{ GOTOm} < replace B>

< test> ::= < IF (logical expression) GOTOn}

< replace a> ::= ({$X_A = X_N$} { $f_A = f_N$})

({ $f_A = f_N$} { $X_A = X_N$})

< replace b> ::= ({$X_B = X_N$} { $f_B = f_N$})

({$f_B = f_N$} { $X_B = X_N$})

If the loop control IF - GOTO statement contains a logical expression that does not reduce to the form of A < relational operator> B it is not acceptable and is rejected.

Within the loop the termination value DELTA should be a constant. We select it from the positional and structural relationships defined in the productions. If logical expression is of the form { X.LT.DELTA} where DELTA is a constant, then we assume we have found the value of DELTA.

If DELTA is a variable we must check the initialize step for an expression of the form { var_A = const.}. Where var_A is DELTA then the constant is the value of DELTA.

If we cannot find the value of DELTA we assume DELTA undefined, and we reduce the plausibility by an appropriate amount.

Next we will check the calculation of X_N by performing a match of statements to see if any are mathematically equivalent to the production using the proper variable names. Then check to see if its followed by the calculation of f_N.

Next check to see if the replacement of old X and f is properly performed by checking for pattern matches with the productions.

Again, if there are failures in the pattern matches or order of operations we again reduce the plausibility by an appropriate step constant.

5. CONCLUSION

The Algorithm Checker is operational for checking implementations of the method of false position algorithm implementations. It has proven to be capable of parsing the implementation into specific steps, isolating specific variables, and making a reasonable guess as to whether a given implementation will in fact perform a false position root finding function.

APPENDIX 1

Implementation Restrictions - method of false position

1. Algorithm Restrictions
   - A. Assume termination within a reasonable number of iterations; therefore there is no need to count the number of steps.
   - B. Assume that the first guess of X_A and X_B will give values of f_A and f_B such that $(f_A * f_B) < 0$.
   - C. Starting values of X_A and X_B will not yield zero values of f_A of f_B.
   - D. The termination conditions are reasonable.
   - E. The termination value, DELTA, is a constant defined within the input program.
2. AC Input Restrictions

   Student programs should observe these constraints:
   - A. The input program must be coded in syntacically legal IBM 360 FORTRAN IV.
   - B. The overall structure of program is that of a FORTRAN subroutine.
   - C. All functions will be defined by function statements.
   - D. All arrays are dimensioned.
   - E. The arithmetic IF is not allowed.
3. Output from AC

   The output from AC will be a printed copy of the input program (above) and comments for AC as to:
   - A. Program variable names and their relation to the algorithm.
   - B. Comments as to coding accuracy.

C. AC's estimate of the plausibility that the correctly implemented.

D. Recommended corrections for the input program.

4. Notation Conventions

X_A, X_B	The two values of X used to calculate the next estimate of the root.
X_N	The new estimate of the root.
f	The function whose root we're interested in.
f(X)	Calculate the function at X.
f_A, f_B	The values of the function at X_A and X_B.

APPENDIX 2

The Algorithm Checker Processing Some Test Implementations of the Method of False Position

```
WELCOME TO THE ALGORITHM CHECKER


* ** ** *   * ** ** *   * ** ** *   * ** ** *   * ** ** *   * ** ** *   * ** ** *   * ** ** *
*START HERE**START HERE**START HERE**START HERE**START HERE**START HERE**START HERE**START HERE*
THIS IS THE INPUT DECK
      SUBROUTINE FALPOS(LEFEND,RITEND,RTFAL,ITFAL)                       00096500
C
C
C     THIS IS AN ALMOST PERFECT PROGRAM
C
C
      REAL LEFEND                                                        00096600
      COMMON A,B,C,D,EPS,MAX                                             00096700
      XL=LEFEND                                                          00096800
      XR=RITEND                                                          00096900
      YL=G(XL)                                                           00097000
      YR=G(XR)                                                           00097100
      RTFAL=XL                                                           00097200
      DO 20 I=1,MAX                                                      00097300
      ITFAL=I                                                            00097400
      RT=RTFAL                                                           00097500
      XM=(XL*YR - XR*YL)/(YR - YL)                                       00097600
      RTFAL=XM                                                           00097700
      YM=G(XM)                                                           00097800
      PRINT,'FALSE POSITION ITERATION ',ITFAL,'ROOT ESTIMATE ',RTFAL     00097900
      IF(ABS(RT - RTFAL).LE. EPS) RETURN                                 00098000
      IF (YM*YL .GT. 0) GOTO 10                                          00098100
      XR=XM                                                              00098200
      YR=YM                                                              00098300
      GOTO 20                                                            00098400
   10 XL=XM                                                              00098500
      YL=YM                                                              00098600
   20 CONTINUE                                                           00098700
      PRINT,'METHOD DID NOT CONVERGE WITHIN ',MAX,'ITERATIONS'           00098800
      RETURN                                                             00098900
      END                                                                00099000
*****************************************************************************************
THE DECK NOW IS:
       SUBROUTINEFALPOS(LEFEND,RITEND,RTFAL,ITFAL)                                                 1
       REALLEFEND                                                                                  2
       COMMONA,B,C,D,EPS,MAX                                                                       3
       XL=LEFEND                                                                                   4
       XR=RITEND                                                                                   5
       YL=G(XL)                                                                                    6
       YR=G(XR)                                                                                    7
       RTFAL=XL                                                                                    8
       DO20I=1,MAX                                                                                 9
       ITFAL=I                                                                                    10
       RT=RTFAL                                                                                   11
       XM=(XL*YR-XR*YL)/(YR-YL)                                                                   12
       RTFAL=XM                                                                                   13
       YM=G(XM)                                                                                   14
       PRINT,'FALSEPOSITIONITERATION',ITFAL,'ROOTESTIMATE',RTFAL                                  15
       IF(ABS(RT-RTFAL).LE.EPS)RETURN                                                             16
       IF(YM*YL.GT.0)GOTO10                                                                       17
       XR=XM                                                                                      18
       YR=YM                                                                                      19
       GOTO20                                                                                     20
    10 XL=XM                                                                                      21
```

```
      YL=YM                                                                                      22
   20 CONTINUE                                                                                   23
      PRINT,'METHODDIDNOTCONVERGEWITHIN',MAX,'ITERATIONS'                                        24
      RETURN                                                                                     25
      END                                                                                        26

AT THIS POINT IN THE ANALYSIS OF SUBROUTINE  ""  THE PLAUSABILITY IS:  100
* ** ** *   LIST VARIABLES PASSED FROM MAIN PROGRAM
THE SUBROUTINE IS NAMED:  FALPOS
THE MAIN PROGRAM PASSED THESE PARAMETERS:
    1     A
    2     B
    3     C
    4     D
    5     EPS
    6     MAX
    7     LEFEND
    8     RITEND
    9     RTFAL
   10     ITFAL
* ** ** *   LEVEL ONE - IS TOP FLOW CORRECT - FIND LOOP STEP
ELIMINATE DO LOOPS, IF ANY, FROM THE INPUT PROGRAM - REPLACE THEM WITH INFINITE LOOPS
THE DO LOOP IS SUPERFLUOUS AND HAS BEEN REMOVED
SUCCESS - MATCH TO PATTERN TYPE TWO
THE DECK NOW IS:
      SUBROUTINEFALPOS(LEFEND,RITEND,RTFAL,ITFAL)                  INITIALIZE   SUBROUTINE STATEMENT  1
      REALLEFEND                                                   INITIALIZE                         2
      COMMONA,B,C,D,EPS,MAX                                        INITIALIZE       COMMON STATEMENT  3
      XL=LEFEND                                                    INITIALIZE                         4
      XR=RITEND                                                    INITIALIZE                         5
      YL=G(XL)                                                     INITIALIZE                         6
      YR=G(XR)                                                     INITIALIZE                         7
      RTFAL=XL                                                     INITIALIZE                         8
   $9 CONTINUE                                                           LOOP                         9
      ITFAL=I                                                            LOOP                        10
      RT=RTFAL                                                           LOOP                        11
      XM=(XL*YR-XR*YL)/(YR-YL)                                           LOOP                        12
      RTFAL=XM                                                           LOOP                        13
      YM=G(XM)                                                           LOOP                        14
      PRINT,'FALSEPOSITIONITERATION',ITFAL,'ROOTESTIMATE',RTFAL          LOOP                        15
      IF(ABS(RT-RTFAL).LT.EPS)GOTO$27                                    LOOP           LOOP CONTROL 16
      IF(YM*YL.GT.0)GOTO10                                               LOOP                        17
      XR=XM                                                              LOOP                        18
      YR=YM                                                              LOOP                        19
      GOTO20                                                             LOOP                        20
   10 XL=XM                                                              LOOP                        21
      YL=YM                                                              LOOP                        22
   20 CONTINUE                                                           LOOP                        23
      GOTO$9                                                             LOOP                        24
      PRINT,'METHODDIDNOTCONVERGEWITHIN',MAX,'ITERATIONS'                LOOP                        25
      RETURN                                                             LOOP                        26
  $27 RETURN                                                        TERMINATE                        27
      END                                                           TERMINATE                        28
AT THIS POINT IN THE ANALYSIS OF SUBROUTINE  "FALPOS"  THE PLAUSABILITY IS:  100
DELTA WAS PASSED TO SUBROUTINE FROM CALLING PROGRAM
DELTA IS OF THE MAGNITUDE  EPS
TERMINATION IS RELATIVE TO ABSOLUTE VALUE OF  RT-RTFAL
AT THIS POINT IN THE ANALYSIS OF SUBROUTINE  "FALPOS"  THE PLAUSABILITY IS:  100
*****************************************************************************************************

* ** ** *   LEVEL TWO - INITIALIZE STEP
* ** ** *   FIND FUNCTION DEFINITION
THE DECK NOW IS:
       SUBROUTINEFALPOS(LEFEND,RITEND,RTFAL,ITFAL)                           INITIALIZE    SUBROUTINE STATEMENT    1
       REALLEFEND                                                            INITIALIZE                            2
       COMMONA,B,C,D,EPS,MAX                                                 INITIALIZE        COMMON STATEMENT    3
       XL=LEFEND                                                             INITIALIZE                            4
       XR=RITEND                                                             INITIALIZE                            5
       YL=G(XL)                                                              INITIALIZE                            6
       YR=G(XR)                                                              INITIALIZE                            7
       RTFAL=XL                                                              INITIALIZE                            8
    $9 CONTINUE                                                                    LOOP                            9
       ITFAL=I                                                                     LOOP                           10
       RT=RTFAL                                                                    LOOP                           11
       XM=(XL*YR-XR*YL)/(YR-YL)                                                    LOOP                           12
       RTFAL=XM                                                                    LOOP                           13
       YM=G(XM)                                                                    LOOP                           14
       PRINT,'FALSEPOSITIONITERATION',ITFAL,'ROOTESTIMATE',RTFAL                   LOOP                           15
       IF(ABS(RT-RTFAL).LT.EPS)GOTO$27                                             LOOP            LOOP CONTROL   16
       IF(YM*YL.GT.0)GOTO10                                                        LOOP                           17
       XR=XM                                                                       LOOP                           18
       YR=YM                                                                       LOOP                           19
       GOTO20                                                                      LOOP                           20
    10 XL=XM                                                                       LOOP                           21
       YL=YM                                                                       LOOP                           22
    20 CONTINUE                                                                    LOOP                           23
       GOTO$9                                                                      LOOP                           24
       PRINT,'METHODDIDNOTCONVERGEWITHIN',MAX,'ITERATIONS'                         LOOP                           25
       RETURN                                                                      LOOP                           26
   $27 RETURN                                                                 TERMINATE                           27
       END                                                                    TERMINATE                           28
* ** ** *   TEST ONE - FIND FUNCTION DEFINITION - ASSUME FUNCTION DEFINED BY FUNCTION STATEMENT
THE FUNCTION WE WANT CANT BE FOUND, EITHER:
          1.  THE FUNCTION IS UNDEFINED
          2.  THE FUNCTION IS EXTERNALLY DEFINED
          3.  THE FUNCTION IS WRITTEN OUT EACH TIME
* ** ** *   TEST TWO - FIND INITIALIZATION OF XA AND XB
AT THIS POINT IN THE ANALYSIS OF SUBROUTINE  "FALPOS"  THE PLAUSABILITY IS:  97

* ** ** *   FIND CALCULATION OF FA AND FB
THE EXPECTED PRECIDENCE IS

     A    B    C    D    E
A    0
B    1    0
C    1    0    0
D    1    1    0    0
E    1    0    1    0    0

THE ACTUAL PRECIDENCE IS:
          DEFINE FUNCTION     STEP A  THIS POSITION IS UNDEFINED
          INITIALIZE XA       STEP B  4
          INITIALIZE XB       STEP C  5
          CALCULATE FA        STEP D  6
          CALCULATE FB        STEP E  7

AT THIS POINT IN THE ANALYSIS OF SUBROUTINE  "FALPOS"  THE PLAUSABILITY IS:  97
*****************************************************************************************************
784
* ** ** *   TEST ONE - FIND CALCULATION OF XN, FIND CALCULATION OF FN FROM XN
(XL*YR-XR*YL)/(YR-YL)
XN IS CALCULATED BY - (XL*YR-XR*YL)/(YR-YL)
```

```
THE DECK NOW IS:
       SUBROUTINEFALPOS(LEFEND,RITEND,RTFAL,ITFAL)                           INITIALIZE     SUBROUTINE STATEMENT    1
       REALLEFEND                                                            INITIALIZE                             2
       COMMONA,B,C,D,EPS,MAX                                                 INITIALIZE         COMMON STATEMENT    3
       XL=LEFEND                                                             INITIALIZE            INITIALIZE XA    4
       XR=RITEND                                                             INITIALIZE            INITIALIZE XB    5
       YL=G(XL)                                                              INITIALIZE             CALCULATE FA    6
       YR=G(XR)                                                              INITIALIZE             CALCULATE FB    7
       RTFAL=XL                                                              INITIALIZE                             8
    $9 CONTINUE                                                                    LOOP                             9
       ITFAL=I                                                                     LOOP                            10
       RT=RTFAL                                                                    LOOP                            11
       XM=(XL*YR-XR*YL)/(YR-YL)                                                    LOOP             CALCULATE XN   12
       RTFAL=XM                                                                    LOOP                            13
       YM=G(XM)                                                                    LOOP             CALCULATE FN   14
       PRINT,'FALSEPOSITIONITERATION',ITFAL,'ROOTESTIMATE',RTFAL                   LOOP                            15
       IF(ABS(RT-RTFAL).LT.EPS)GOTO$27                                             LOOP             LOOP CONTROL   16
       IF(YM*YL.GT.0)GOTO10                                                        LOOP                            17
       XR=XM                                                                       LOOP                            18
       YR=YM                                                                       LOOP                            19
       GOTO20                                                                      LOOP                            20
    10 XL=XM                                                                       LOOP                            21
       YL=YM                                                                       LOOP                            22
    20 CONTINUE                                                                    LOOP                            23
       GOTO$9                                                                      LOOP                            24
       PRINT,'METHODDIDNOTCONVERGEWITHIN',MAX,'ITERATIONS'                         LOOP                            25
       RETURN                                                                      LOOP                            26
   $27 RETURN                                                                 TERMINATE                            27
       END                                                                    TERMINATE                            28
XA = XL
THE INITIAL VALUE OF XA IS - LEFEND
XB = XR
THE INITIAL VALUE OF XB IS - RITEND
FA = YL
FB = YR
XN = XM
XN IS CALCULATED FROM THE EXPRESSION:  (XL*YR-XR*YL)/(YR-YL)
FN = YM
FUNCTION = G
AT THIS POINT IN THE ANALYSIS OF SUBROUTINE  "FALPOS"  THE PLAUSABILITY IS:  97
* ** ** *   FIND SELECTION OF NEW X AND ASSOCIATED F
WE HAVE PERFORMED A SUCCESSFUL PATTERN MATCH TO SELECTION PATTERN
          SECTION ONE
               XR = XM           18
               YR = YM           19
          SECTION TWO
               XL = XM           21
               YL = YM           22
SUCCESS - MATCH TO TX2
THE DECK NOW IS:
       SUBROUTINEFALPOS(LEFEND,RITEND,RTFAL,ITFAL)                           INITIALIZE     SUBROUTINE STATEMENT    1
       REALLEFEND                                                            INITIALIZE                             2
       COMMONA,B,C,D,EPS,MAX                                                 INITIALIZE         COMMON STATEMENT    3
       XL=LEFEND                                                             INITIALIZE            INITIALIZE XA    4
       XR=RITEND                                                             INITIALIZE            INITIALIZE XB    5
       YL=G(XL)                                                              INITIALIZE             CALCULATE FA    6
       YR=G(XR)                                                              INITIALIZE             CALCULATE FB    7
       RTFAL=XL                                                              INITIALIZE                             8
    $9 CONTINUE                                                                    LOOP                             9
       ITFAL=I                                                                     LOOP                            10
       RT=RTFAL                                                                    LOOP                            11
```

```
      XM=(XL*YR-XR*YL)/(YR-YL)                                              LOOP         CALCULATE XN   12
      RTFAL=XM                                                              LOOP                        13
      YM=G(XM)                                                              LOOP         CALCULATE FN   14
      PRINT,'FALSEPCSITIONITERATICN',ITFAL,'ROOTESTIMATE',RTFAL             LOOP                        15
      IF(ABS(RT-RTFAL).LT.EPS)GOTO$27                                       LOOP         LOOP CONTROL   16
      IF(YM*YL.GT.0)GOTO10                                                  LOOP    SELECTION CONTROL   17
      XR=XM                                                                 LOOP    RECALC XB FROM XN   18
      YR=YM                                                                 LOOP    RECALC FB FROM FN   19
      GOTC20                                                                LOOP                        20
   10 XL=XM                                                                 LOOP    RECALC XA FROM XN   21
      YL=YM                                                                 LOOP    RECALC FA FROM FN   22
   20 CCNTINUE                                                              LOOP                        23
      GOTC$9                                                                LOOP                        24
      PRINT,'METHODDIDNOTCCNVERGEWITHIN',MAX,'ITERATIONS'                   LOOP                        25
      RETURN                                                                LOOP                        26
  $27 RETURN                                                           TERMINATE                        27
      END                                                              TERMINATE                        28
XA = XL
THE INITIAL VALUE OF XA IS - LEFEND
XB = XR
THE INITIAL VALUE OF XB IS - RITEND
FA = YL
FB = YR
XN = XM
XN IS CALCULATED FRCM THE EXPRESSION:  (XL*YR-XR*YL)/(YR-YL)
FN = YM
FUNCTION = G
AT THIS POINT IN THE ANALYSIS OF SUBROUTINE  "FALPOS"  THE PLAUSABILITY IS:  97
THE EXPECTED LCOP PRECEDENCE IS:

CALCULATE XN             STEP A  1
CALCULATE FN FROM XN     STEP B  2
LOOP CONTROL             STEP C  3
SELECT NEW XA            STEP D  4  OR  SELECT NEW XA          STEP D  6
SELECT NEW FA            STEP E  5  OR  SELECT NEW FA          STEP E  7
SELECT NEW XB            STEP F  6  OR  SELECT NEW XB          STEP F  4
SELECT NEW FB            STEP G  7  OR  SELECT NEW FB          STEP G  5

THE ACTUAL LOCP PRECEDENCE IS
CALCULATE XN             STEP A  12
CALCULATE FN FROM XN     STEP B  14
LOOP CCNTROL             STEP C  16
SELECT NEW XA            STEP D  21
SELECT NEW FA            STEP E  22
SELECT NEW XB            STEP F  18
SELECT NEW FB            STEP G  19
THE FINAL PLAUSIBILITY OF THIS MECHANIZATICN IS:  97


* ** ** *   * ** ** *   * ** ** *   * ** ** *   * ** ** *   * ** ** *   * ** ** *   * ** ** *
*START HERE**START HERE**START HERE**START HERE**START HERE**START HERE**START HERE**START HERE*
THIS IS THE INPUT DECK
      SUBROUTINEFALSEP(XLO,XRC,NIT7X,EPS,FALSP)                             00105308
C
C
C     THIS PROGRAM HAS A FEW FLAWS
C
C
      FMN(X) = F(X)                                                         00105312
      XL = XLO                                                              00105316
      XR = XRO                                                              00105320
      NIT = 0                                                               00105324
      YL = FMN(XL)                                                          00105328
      YR = FMN(XR)                                                          00105332
10    XM = X                                                                00105336
10    XM = (XL * YR - XR - YL)/(YR - YL)                                    00105340
      YM = FMN(XM)                                                          00105344
      NIT = NIT + 1                                                         00105348
      IF (YM * YL .LE.0) GO TO 12                                           00105352
      XL = XM                                                               00105356
      YL = YM                                                               00105360
      IF((ABS(XL - XR).LE.EFS) .CR. (NIT.GE.NITMX)) GO TO 14                00105362
      GO TC 10                                                              00105364
12    XR = XM                                                               00105368
      YR = YM                                                               00105372
      IF((ABS(XL - XR).LE.EPS) .OR. (NIT.GF.NITMX)) GO TO 14                00105376
14    PRINT,'RCOT',XM,'NIT=',NIT                                            00105380
      RETURN                                                                00105384
      END                                                                   00105388
******************************************************************************************
THE DECK NCW IS:
        SUBRCUTINEFALSEP(XLO,XRO,NIT7X,EPS,FALSP)                                                1
        FMN(X)=F(X)                                                                              2
        XL=XLO                                                                                   3
        XR=XRO                                                                                   4
        NIT=0                                                                                    5
        YL=FMN(XL)                                                                               6
        YR=FMN(XR)                                                                               7
     10 XM=X                                                                                     8
     10 XM=(XL*YR-XR-YL)/(YR-YL)                                                                 9
        YM=FMN(XM)                                                                              10
        NIT=NIT+1                                                                               11
        IF(YM*YL.LE.0)GOTO12                                                                    12
        XL=XM                                                                                   13
        YL=YM                                                                                   14
        IF((ABS(XL-XR).LE.EPS).OR.(NIT.GE.NITMX))GOTO14                                         15
        GCTO10                                                                                  16
     12 XR=XM                                                                                   17
        YR=YM                                                                                   18
        IF((ABS(XL-XR).LE.EPS).CR.(NIT.GE.NITMX))GOTO14                                         19
     14 PRINT,'ROOT',XM,'NIT=',NIT                                                              20
        RETURN                                                                                  21
        END                                                                                     22

AT THIS POINT IN THE ANALYSIS OF SUBROUTINE  "FALPOS"  THE PLAUSABILITY IS:  100
* ** ** *   LIST VARIABLES PASSED FRCM MAIN PROGRAM
THE SUBROUTINE IS NAMED:  FALSEP
THE MAIN PROGRAM PASSED THESE PARAMETERS:
     1   XLO
     2   XRO
     3   NIT7X
     4   EPS
     5   FALSP
* ** ** *   LEVEL CNE - IS TCP FLOW CCRRECT - FIND LOOP STEP
ELIMINATE DO LCOPS, IF ANY, FRCM THE INPUT PROGRAM - REPLACE THEM WITH INFINITE LOOPS
SUCCESS - MATCH TO PATTERN TYPE TWO
THE DECK NCW IS:
        SUBROUTINEFALSEP(XLO,XRO,NIT7X,EPS,FALSP)                       INITIALIZE     SUBROUTINE STATEMENT    1
        FMN(X)=F(X)                                                     INITIALIZE                             2
        XL=XLO                                                          INITIALIZE                             3
        XR=XRO                                                          INITIALIZE                             4
        NIT=0                                                           INITIALIZE                             5
      YL=FMN(XL)                                                           INITIALIZE                               6
      YR=FMN(XR)                                                           INITIALIZE                               7
   10 XM=X                                                                 INITIALIZE                               8
   10 XM=(XL*YR-XR-YL)/(YR-YL)                                                   LOOP                               9
      YM=FMN(XM)                                                                 LOOP                              10
      NIT=NIT+1                                                                  LOOP                              11
      IF(YM*YL.LT.0)GOTO12                                                       LOOP             LOOP CONTROL     12
      XL=XM                                                                      LOOP                              13
      YL=YM                                                                      LOOP                              14
      IF((ABS(XL-XR).LE.EPS).OR.(NIT.GE.NITMX))GOTO14                            LOOP                              15
      GOTO10                                                                     LOOP                              16
   12 XR=XM                                                                 TERMINATE                              17
      YR=YM                                                                 TERMINATE                              18
      IF((ABS(XL-XR).LE.EPS).OR.(NIT.GE.NITMX))GOTO14                       TERMINATE                              19
   14 PRINT,'ROOT',XM,'NIT=',NIT                                            TERMINATE                              20
      RETURN                                                                TERMINATE                              21
      END                                                                   TERMINATE                              22
AT THIS POINT IN THE ANALYSIS OF SUBRCUTINE  "FALSEP"  THE PLAUSABILITY IS:  100
FIRST GUESS AT DELTA IS ZERO - REWORK LOGICAL EXPRESSION
LOGICAL EXPRESSION DOES NOT MATCH AN ACCEPTABLE PATTERN
MAGNITUDE AND NAME OF DELTA UNDEFINED
AT THIS POINT IN THE ANALYSIS OF SUBROUTINE  "FALSEP"  THE PLAUSABILITY IS:  89
*****************************************************************************************************

* ** ** *   LEVEL TWO - INITIALIZE STEP
* ** ** *   FIND FUNCTION DEFINITION
THE DECK NOW IS:
      SUBROUTINEFALSEP(XLO,XRO,NIT7X,EPS,FALSP)                            INITIALIZE     SUBROUTINE STATEMENT      1
      FMN(X)=F(X)                                                          INITIALIZE                               2
      XL=XLO                                                               INITIALIZE                               3
      XR=XRO                                                               INITIALIZE                               4
      NIT=0                                                                INITIALIZE                               5
      YL=FMN(XL)                                                           INITIALIZE                               6
      YR=FMN(XR)                                                           INITIALIZE                               7
   10 XM=X                                                                 INITIALIZE                               8
   10 XM=(XL*YR-XR-YL)/(YR-YL)                                                   LOOP                               9
      YM=FMN(XM)                                                                 LOOP                              10
      NIT=NIT+1                                                                  LOOP                              11
      IF(YM*YL.LT.0)GOTO12                                                       LOOP             LOOP CONTROL     12
      XL=XM                                                                      LOOP                              13
      YL=YM                                                                      LOOP                              14
      IF((ABS(XL-XR).LE.EPS).OR.(NIT.GE.NITMX))GOTO14                            LOOP                              15
      GOTO10                                                                     LOOP                              16
   12 XR=XM                                                                 TERMINATE                              17
      YR=YM                                                                 TERMINATE                              18
      IF((ABS(XL-XR).LE.EPS).OR.(NIT.GE.NITMX))GOTO14                       TERMINATE                              19
   14 PRINT,'ROOT',XM,'NIT=',NIT                                            TERMINATE                              20
      RETURN                                                                TERMINATE                              21
      END                                                                   TERMINATE                              22
* ** ** *   TEST ONE - FIND FUNCTION DEFINITION - ASSUME FUNCTION DEFINED BY FUNCTION STATEMENT
FUNCTION = FMN
THE FUNCTION REPRESINTS THE EXPRESSION:  F(X)
* ** ** *   TEST TWO - FIND INITIALIZATION OF XA AND XB
AT THIS POINT IN THE ANALYSIS OF SUBROUTINE  "FALSEP"  THE PLAUSABILITY IS:  89

* ** ** *   FIND CALCULATION OF FA AND FB
THE EXPECTED PRECIDENCE IS

     A    B    C    D    E
A    0
B    1    0
```

```
C    1    0    0
D    1    1    0    0
E    1    0    1    0    0

THE ACTUAL PRECIDENCE IS:
          DEFINE FUNCTION      STEP A  2
          INITIALIZE XA        STEP B  3
          INITIALIZE XB        STEP C  4
          CALCULATE FA         STEP D  6
          CALCULATE FB         STEP E  7

AT THIS POINT IN THE ANALYSIS OF SUBROUTINE  "FALSEP"  THE PLAUSABILITY IS:  87
****************************************************************************************
1900
* ** ** *   TEST ONE - FIND CALCULATION OF XN, FIND CALCULATION OF FN FROM XN
(XL*YR-XR-YL)/(YR-YL)
XN IS CALCULATED BY - (XL*YR-XR-YL)/(YR-YL)
THE DECK NOW IS:
       SUBROUTINEFALSEP(XLO,XRO,NIT7X,EPS,FALSP)                              INITIALIZE     SUBROUTINE STATEMENT    1
       FMN(X)=F(X)                                                            INITIALIZE         FUNCTION DEFINED    2
       XL=XLO                                                                 INITIALIZE            INITIALIZE XA    3
       XR=XRO                                                                 INITIALIZE            INITIALIZE XB    4
       NIT=0                                                                  INITIALIZE                             5
       YL=FMN(XL)                                                             INITIALIZE             CALCULATE FA    6
       YR=FMN(XR)                                                             INITIALIZE             CALCULATE FB    7
    10 XM=X                                                                   INITIALIZE                             8
    10 XM=(XL*YR-XR-YL)/(YR-YL)                                                     LOOP             CALCULATE XN    9
       YM=FMN(XM)                                                                   LOOP             CALCULATE FN   10
       NIT=NIT+1                                                                    LOOP                            11
       IF(YM*YL.LT.0)GOTO12                                                         LOOP             LOOP CONTROL   12
       XL=XM                                                                        LOOP                            13
       YL=YM                                                                        LOOP                            14
       IF((ABS(XL-XR).LE.EPS).OR.(NIT.GE.NITMX))GOTO14                              LOOP                            15
       GOTO10                                                                       LOOP                            16
    12 XR=XM                                                                   TERMINATE                            17
       YR=YM                                                                   TERMINATE                            18
       IF((ABS(XL-XR).LE.EPS).OR.(NIT.GE.NITMX))GOTO14                         TERMINATE                            19
    14 PRINT,'ROOT',XM,'NIT=',NIT                                              TERMINATE                            20
       RETURN                                                                  TERMINATE                            21
       END                                                                     TERMINATE                            22
XA = XL
THE INITIAL VALUE OF XA IS - XLO
XB = XR
THE INITIAL VALUE OF XB IS - XRO
FA = YL
FB = YR
XN = XM
XN IS CALCULATED FROM THE EXPRESSION:  (XL*YR-XR-YL)/(YR-YL)
FN = YM
FUNCTION = FMN
THE FUNCTION REPRESENTS THE EXPRESSION: . F(X)
AT THIS POINT IN THE ANALYSIS OF SUBROUTINE  "FALSEP"  THE PLAUSABILITY IS:  87
* ** ** *   FIND SELECTION OF NEW X AND ASSOCIATED F
FAILED TO FIND SELECTION PATTERN FOR NEW X & F
THE DECK NOW IS:
       SUBROUTINEFALSEP(XLO,XRO,NIT7X,EPS,FALSP)                              INITIALIZE     SUBROUTINE STATEMENT    1
       FMN(X)=F(X)                                                            INITIALIZE         FUNCTION DEFINED    2
       XL=XLO                                                                 INITIALIZE            INITIALIZE XA    3
       XR=XRO                                                                 INITIALIZE            INITIALIZE XB    4
       NIT=0                                                                  INITIALIZE                             5
       YL=FMN(XL)                                                             INITIALIZE             CALCULATE FA    6
       YR=FMN(XR)                                                             INITIALIZE            CALCULATE FB     7
    10 XM=X                                                                   INITIALIZE                             8
    10 XM=(XL*YR-XR-YL)/(YR-YL)                                                     LOOP            CALCULATE XN     9
       YM=FMN(XM)                                                                   LOOP            CALCULATE FN    10
       NIT=NIT+1                                                                    LOOP                            11
       IF(YM*YL.LT.0)GOTO12                                                         LOOP            LOOP CONTROL    12
       XL=XM                                                                        LOOP                            13
       YL=YM                                                                        LOOP                            14
       IF((ABS(XL-XR).LE.EPS).OR.(NIT.GE.NITMX))GOTO14                              LOOP                            15
       GOTO10                                                                       LOOP                            16
    12 XR=XM                                                                   TERMINATE                            17
       YR=YM                                                                   TERMINATE                            18
       IF((ABS(XL-XR).LE.EPS).OR.(NIT.GE.NITMX))GOTO14                         TERMINATE                            19
    14 PRINT,'ROOT',XM,'NIT=',NIT                                              TERMINATE                            20
       RETURN                                                                  TERMINATE                            21
       END                                                                     TERMINATE                            22
XA = XL
THE INITIAL VALUE OF XA IS - XLO
XB = XR
THE INITIAL VALUE OF XB IS - XRO
FA = YL
FB = YR
XN = XM
XN IS CALCULATED FROM THE EXPRESSION:  (XL*YR-XR-YL)/(YR-YL)
FN = YM
FUNCTION = FMN
THE FUNCTION REPRES)NTS THE EXPRESSION:  F(X)
AT THIS POINT IN THE ANALYSIS OF SUBROUTINE  "FALSEP"  THE PLAUSABILITY IS:  69
THE EXPECTED LOOP PRECEDENCE IS:

CALCULATE XN             STEP A  1
CALCULATE FN FROM XN     STEP B  2
LOOP CONTROL             STEP C  3
SELECT NEW XA            STEP D  4  OR  SELECT NEW XA            STEP D  6
SELECT NEW FA            STEP E  5  OR  SELECT NEW FA            STEP E  7
SELECT NEW XB            STEP F  6  OR  SELECT NEW XB            STEP F  4
SELECT NEW FB            STEP G  7  OR  SELECT NEW FB            STEP G  5

THE ACTUAL LOOP PRECEDENCE IS
CALCULATE XN             STEP A  9
CALCULATE FN FROM XN     STEP B  10
LOOP CONTROL             STEP C  12
SELECT NEW XA            STEP D  THIS POSITION IS UNDEFINED
SELECT NEW FA            STEP E  THIS POSITION IS UNDEFINED
SELECT NEW XB            STEP F  THIS POSITION IS UNDEFINED
SELECT NEW FB            STEP G  19
THE FINAL PLAUSIBILITY OF THIS MECHANIZATION IS:  69


* ** ** *   * ** ** *   * ** ** *   * ** ** *   * ** ** *   * ** ** *   * ** ** *   * ** ** *
*START HERE**START HERE**START HERE**START HERE**START HERE**START HERE**START HERE**START HERE*
THIS IS THE INPUT DECK
              SUBROUTINE CON (A,B,C,D)                               00106500
C
C
C     THIS PROGRAM IS POOR
C
C
             F(X) = X**2 + Y+Z                                       00106600
             TEST = .050                                             00106700
50           NEW = F(Y)                                              00106800
```

```
            Q = ABS(NEW)                                                            00106900
            IF(0.GT.TEST-Q) GO TO 50                                                00107000
            STOP                                                                    00107100
            END                                                                     00107200
*************************************************************************************************
THE DECK NOW IS:
      SUBROUTINEDCN(A,B,C,D)                                                                                        1
      F(X)=X**2+Y+Z                                                                                                 2
      TEST=.050                                                                                                     3
   50 NEW=F(Y)                                                                                                      4
      Q=ABS(NEW)                                                                                                    5
      IF(0.GT.TEST-Q)GOTO50                                                                                         6
      STOP                                                                                                          7
      END                                                                                                           8

AT THIS POINT IN THE ANALYSIS OF SUBROUTINE  "FALSEP"  THE PLAUSABILITY IS:  100
* ** ** *   LIST VARIABLES PASSED FROM MAIN PROGRAM
THE SUBROUTINE IS NAMED:  DCN
THE MAIN PROGRAM PASSED THESE PARAMETERS:
    1      A
    2      B
    3      C
    4      D
* ** ** *   LEVEL ONE - IS TOP FLOW CORRECT - FIND LOOP STEP
ELIMINATE DO LOOPS, IF ANY, FROM THE INPUT PROGRAM - REPLACE THEM WITH INFINITE LOOPS
SUCCESS - MATCH TO PATTERN TYPE ONE
THE DECK NOW IS:
      SUBROUTINEDCN(A,B,C,D)                                              INITIALIZE     SUBROUTINE STATEMENT    1
      F(X)=X**2+Y+Z                                                       INITIALIZE                             2
      TEST=.050                                                           INITIALIZE                             3
   50 NEW=F(Y)                                                                  LOOP                             4
      Q=ABS(NEW)                                                                LOOP                             5
      IF(0.GT.TEST-Q)GOTO50                                                     LOOP           LOOP CONTROL      6
      STOP                                                                 TERMINATE                             7
      END                                                                  TERMINATE                             8
AT THIS POINT IN THE ANALYSIS OF SUBROUTINE  "DON"  THE PLAUSABILITY IS:  100
LEFT HAND EXPRESSION IS ZERO - REWORK LOGICAL EXPRESSION
DELTA IS OF THE MAGNITUDE   .050
TERMINATION IS RELATIVE TO ABSOLUTE VALUE OF  NEW
AT THIS POINT IN THE ANALYSIS OF SUBROUTINE  "DON"  THE PLAUSABILITY IS:  100
*************************************************************************************************

* ** ** *   LEVEL TWO - INITIALIZE STEP
* ** ** *   FIND FUNCTION DEFINITION
THE DECK NOW IS:
      SUBROUTINEDCN(A,B,C,D)                                              INITIALIZE     SUBROUTINE STATEMENT    1
      F(X)=X**2+Y+Z                                                       INITIALIZE                             2
      TEST=.050                                                           INITIALIZE            DELTA DEFINED    3
   50 NEW=F(Y)                                                                  LOOP                             4
      Q=ABS(NEW)                                                                LOOP                             5
      IF(Q.GT.TEST)GOTO50                                                       LOOP           LOOP CONTROL      6
      STOP                                                                 TERMINATE                             7
      END                                                                  TERMINATE                             8
* ** ** *   TEST ONE - FIND FUNCTION DEFINITION - ASSUME FUNCTION DEFINED BY FUNCTION STATEMENT
FUNCTION = F
THE FUNCTION REPRESENTS THE EXPRESSION:   X**2+Y+Z
* ** ** *   TEST TWO - FIND INITIALIZATION OF XA AND XB
NEITHER XA OR XB WAS INITIILIZED BY SUBROUTINE
AT THIS POINT IN THE ANALYSIS OF SUBROUTINE  "DON"  THE PLAUSABILITY IS:  80

* ** ** *   FIND CALCULATION OF FA AND FB
```

```
THE EXPECTED PRECIDENCE IS

     A    B    C    D    E
A    0
B    1    0
C    1    0    0
D    1    1    0    0
E    1    0    1    0    0

THE ACTUAL PRECIDENCE IS:
           DEFINE FUNCTION      STEP A  2
           INITIALIZE XA        STEP B  THIS POSITION IS UNDEFINED
           INITIALIZE XB        STEP C  THIS POSITION IS UNDEFINED
           CALCULATE FA         STEP D  THIS POSITION IS UNDEFINED
           CALCULATE FB         STEP E  THIS POSITION IS UNDEFINED

AT THIS POINT IN THE ANALYSIS OF SUBROUTINE  "DON"  THE PLAUSABILITY IS:  80
************************************************************************************************
2717
* ** ** *   TEST ONE - FIND CALCULATICN OF XN, FIND CALCULATION OF FN FROM XN
FAILED TO FIND CALCULATICN CF EITHER FN OR XN
AT THIS POINT IN THE ANALYSIS CF SUBROUTINE  "DON"  THE PLAUSABILITY IS:  64
* ** ** *   FIND SELECTION CF NEW X AND ASSOCIATED F
FAILED TO FIND SELECTICN PATTERN FOR NEW X & F
THE DECK NOW IS:
       SUBROUTINEDON(A,B,C,C)                                                  INITIALIZE     SUBROUTINE STATEMENT    1
       F(X)=X**2+Y+Z                                                           INITIALIZE         FUNCTION DEFINED    2
       TEST=.050                                                               INITIALIZE            DELTA DEFINED    3
    50 NEW=F(Y)                                                                      LOOP                             4
       Q=ABS(NEW)                                                                    LOOP                             5
       IF(Q.GT.TEST)GOTO50                                                           LOOP             LOOP CONTROL    6
       STOP                                                                     TERMINATE                             7
       END                                                                      TERMINATE                             8
XA = Y
FUNCTICN = F
THE FUNCTICN REPRESENTS THE EXPRESSION:  X**2+Y+Z
AT THIS POINT IN THE ANALYSIS OF SUBROUTINE  "DON"  THE PLAUSABILITY IS:  51
THE EXPECTED LCCP PRECEDENCE IS:

CALCULATE XN              STEP A  1
CALCULATE FN FROM XN      STEP B  2
LOOP CONTROL              STEP C  3
SELECT NEW XA             STEP D  4  OR  SELECT NEW XA           STEP D  6
SELECT NEW FA             STEP E  5  OR  SELECT NEW FA           STEP E  7
SELECT NEW XB             STEP F  6  OR  SELECT NEW XB           STEP F  4
SELECT NEW FB             STEP G  7  OR  SELECT NEW FB           STEP G  5

THE ACTUAL LOOP PRECEDENCE IS
CALCULATE XN              STEP A  THIS POSITION IS UNDEFINED
CALCULATE FN FROM XN      STEP B  THIS POSITION IS UNDEFINED
LOCP CCNTROL              STEP C  6
SELECT NEW XA             STEP D  THIS POSITION IS UNDEFINED
SELECT NEW FA             STEP E  THIS POSITION IS UNDEFINED
SELECT NEW XB             STEP F  THIS POSITION IS UNDEFINED
SELECT NEW FB             STEP G  19
THE FINAL PLAUSIBILITY OF THIS MECHANIZATICN IS:  51


* ** ** *   * ** ** *   * ** ** *   * ** ** *   * ** ** *   * ** ** *   * ** ** *   * ** ** *
*START HERE**START HERE**START HERE**START HERE**START HERE**START HERE**START HERE**START HERE*
THIS IS THE INPUT DECK
```

```
NORMAL TERMINATION IN STATEMENT        161

NUMBER OF STATEMENTS EXECUTED         4298

EXECUTION TIME (SECONDS)             2.817

MSEC/STATEMENT                       0.655

NUMBER OF STORAGE REGENERATIONS          3
```

APPENDIX 3

The Algorithm Checker Program

```
S P I T B O L -- VERSION 2.2 -- COPY NUMBER 2023

ILLINOIS INSTITUTE OF TECHNOLOGY

SYSTEM = OS/360


PARAMETER VALUES FOR THIS RUN

L = 16384
H = 1024000
R = 8192
C = 100000
P = 100000
T = 55
D = 10
N = 58
I = 0


DYNAMIC MEMORY AVAILABLE = 174080


TF APPLIED 2.2.1
TF APPLIED 2.2.2
TF APPLIED 2.2.3
TF APPLIED 2.2.4
```

```
************************************************************************ .00000700
************* VARIABLE NAMES USED IN MODEL           ******************* .00000800
*************                                        ******************* .00000900
************* DELTA - ITERATIVE TERMINATION VALUE    ******************* .00001000
************* XA - X VALUE ASSOCIATED WITH POSITIVE Y******************* .00001100
************* XN - X ASSOCIATED WITH NEW Y            ******************* .00001200
************* XB - X VALUE ASSOCIATED WITH NEGATIVE Y******************* .00001300
************* FA - POSITIVE Y                         ******************* .00001400
************* FB - NEGATIVE Y                         ******************* .00001500
*************                                        ******************* .00001600
************* FN - NEW Y                              ******************* .00001700
************************************************************************ .00001800
             PROB.A = 0.01     ;* LOOP EXISTANCE                         .00001900
             PROB.B = 0.95     ;* DELTA IS A VARIABLE - CAN' FIND IT     .00002000
             PROB.C = 0.85     ;* CAN'T FIND DELTA                       .00002100
             PROB.D = 0.90     ;* CAN'T UNSNARL TERMINATION EXP          .00002200
             PROB.E = 0.95     ;* TERMINATION EXPR IS NOT NECESSARILY PO .00002300
             PROB.F = 0.97                                               .00002400
             PROB.G = 0.80                                               .00002500
             PROB.H = 0.80                                               .00002600
             PROB.I = 0.80                                               .00002700
             PROB.J = 0.90                                               .00002800
             PROB.K = 0.97                                               .00002900
             PROB.L = 0.80                                               .00003000
             PROB.M = 0.80                                               .00003100
             PROB.N = 0.65                                               .00003200
             PROB.P = 0.9                                                .00003300
             PROB.Q = 0.9                                                .00003400
             PROB.R = 0.50                                               .00003500
             PROB.S = 0.95                                               .00003600
             PROB.T = 0.9                                                .00003700
             PROB.U = 0.95                                               .00003800
             PROB.V = 0.80     ;* FAILED SELECTION PATTERN               .00003850
             NULL =                                                      .00003900
             &ANCHOR = 0                                                 .00004000
             &FULLSCAN = 1                                               .00004100
             &TRIM = 1                                                   .00004200
             DEFINE('TIMEOUT()')                                         .00004300
             DEFINE('REPLACE(STRING,IN,OUT)TEMP1,TEMP2,TEMP3')           .00004400
             DEFINE('SCAN(ARAY,ELEM,N.POS)')                             .00004500
             DEFINE('SCAN.VAR(XLL)LL')                                   .00004600
             DEFINE('SCAN.PARM(PARM.X)')                                 .00004700
             DEFINE('CONFLICT(C.X,C.Y,C.Z)')                             .00004800
             DEFINE('VAROUT()')                                          .00004900
             DEFINE('DECKOUT()W')                                        .00005000
             DEFINE('FZERO(X)')                                          .00005100
             DEFINE('REAL(X)')                                           .00005200
             DEFINE('VARIABLE(X)Y,Z')                                    .00005300
             DEFINE('PROBOUT()')                                         .00005400
             COMMENT = POS(0) 'C'                                        .00005500
             FIELD = POS(0) BREAK(' ') . STR SPAN(' ')                   .00005600
             ABSF = POS(0) 'ABS(' BAL . YCT ')' RPOS(0)                  .00005700
             ALPHA = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'                        .00005800
             DIGIT = '1234567890'                                        .00005900
             XINT = ANY(DIGIT) (SPAN(DIGIT) | NULL)                      .00006000
             UNARY = '+' | '-'                                           .00006100
```

```
45              XREAL = XINT '.' (SPAN(DIGIT) | NULL)                        .00006200
    +           | (SPAN(DIGIT) | NULL) '.' XINT                              .00006300
46              INT = XINT | XINT 'F' (UNARY | NULL) XINT                    .00006400
47              ALPHAMERIC = ALPHA DIGIT                                     .00006500
48              T.OP = ANY('+-*/=()')                                        .00006600
49              XEXP = ('D' | 'E') (UNARY | NULL) XREAL                      .00006700
50              REAL = XREAL (XEXP | NULL)                                   .00006800
51              NUMBER = INT | REAL                                          .00006900
52              SIGN.NUMBER = NUMBER | UNARY NUMBER                          .00007000
53              VAR = (ANY(ALPHA) (SPAN(ALPHAMERIC) | NULL))                 .00007100
54              ID = VAR | NUMBER                                            .00007200
55              LESS = '.LT.' | '.LE.'                                       .00007300
56              GREATER = '.GT.' | '.GE.'                                    .00007400
57              LOGOP = LESS | GREATER                                       .00007500
58              M.OP = '**' | '+' | '-' | '*' | '/'                          .00007600
59              LLEXP = (POS(0) BAL . EXP2 LESS BAL . EXP1 RPOS(0))          .00007700
60              GLEXP2 = (POS(0) BAL . EXP2 GREATER BAL . EXP1 RPOS(0))      .00007800
61              LLEXP2 = (POS(0) BAL . EXP1 LESS BAL . EXP2 RPOS(0))         .00007900
62              GLEXP = (POS(0) BAL . EXP1 GREATER BAL . EXP2 RPOS(0))       .00008000
63              LGXP = (BAL . EXP1 LOGOP . OP BAL . EXP2)                    .00008100
    +           | ('.NOT.(' *LGXP ')')                                       .00008200
64              LGXP2 = POS(0) LGXP RPOS(0)                                  .00008300
65              GOTOST = POS(0) 'GOTO' (INT | '$' INT) . XST RPOS(0)         .00008400
66              IFGOTO =  POS(0) 'IF(' LGXP . LOGEXP ')GOTO'                 .00008500
    +           (XINT | '$' XINT) . XST RPOS(0)                              .00008600
67              IFRETURN = POS(0) 'IF(' LGXP . LOGEXP ')RETURN' RPOS(0)      .00008700
68              VAR.ST = ID |  ID ','  *VAR.ST                               .00008800
69              SIMPLEAS = (VAR . ZL '=' VAR . ZR)                           .00008900
    +           . SIMPLE.AS.OUT                                              .00009000
70              IFST = (POS(0) 'IF(' LGXP . LOGEXP ')'                       .00009100
    +           SIMPLEAS RPOS(0))                                            .00009200
71              SIMPLEAS2 = POS(0) SIMPLEAS RPOS(0)                          .00009300
72              ZERO = (SPAN('0') (('.' (SPAN('0') | NULL)) | NULL))         .00009400
73              SIMPLE.AS.N = VAR . X '=' (SIGN.NUMBER | VAR) . XR           .00009500
74              SIMPLE.AS.N2 = POS(0) SIMPLE.AS.N RPOS(0)                    .00009600
75              EXP = *FACT | *FACT '+' *EXP | *FACT '-' *EXP                .00009700
76              FACT = *PROD | *PROD '*' *FACT | *PROD '/' *FACT             .00009800
77              PROD = *POWER | *POWER '**' *PROD                            .00009900
78              POWER = (ID | ID '(' VAR.ST ')' | '(' *EXP ')')              .00010000
79              PVV = ('ABS(' BAL ')' | 'ABS(' BAL ')' M.OP *PVV             .00010100
    +           | NUMBER M.OP *PVV)                                          .00010200
80              PVV2 = POS(0) PVV RPOS(0)                                    .00010300
81              AS.ST = VAR '=' EXP                                          .00010400
82              DO.STATEMENT = POS(0) 'DO' XINT  . XSTN VAR '='              .00010500
    +           (VAR | XINT) ',' (VAR | XINT) (',' (VAR | XINT) | NULL)      .00010600
    +           RPOS(0)                                                      .00010700
83              AS.ST2 = POS(0) AS.ST RPOS(0)                                .00010800
   ************* DEFINE OUTPUT STATEMENTS                ******************* .00010900
84              X.STEP.A1 = 'DEFINE FUNCTION'                                .00011000
85              X.STEP.B1 = 'INITIALIZE XA'                                  .00011100
86              X.STEP.C1 = 'INITIALIZE XB'                                  .00011200
87              X.STEP.D1 =  'CALCULATE FA'                                  .00011300
88              X.STEP.E1 = 'CALCULATE FB'                                   .00011400
89              X.STEP.A = X.STEP.A1 DUPL(' ',20 - SIZE(X.STEP.A1))          .00011500
    +           'STEP A  '                                                   .00011600
90              X.STEP.B = X.STEP.B1 DUPL(' ',20 - SIZE(X.STEP.B1))          .00011700
```

```
    +           'STEP B  '                                                 .00011800
 91             X.STEP.C = X.STEP.C1 DUPL(' ',20 - SIZE(X.STEP.C1))        .00011900
    +           'STEP C  '                                                 .00012000
 92             X.STEP.D = X.STEP.D1 DUPL(' ',20 - SIZE(X.STEP.D1))        .00012100
    +           'STEP D  '                                                 .00012200
 93             X.STEP.E = X.STEP.E1 DUPL(' ',20 - SIZE(X.STEP.E1))        .00012300
    +           'STEP E  '                                                 .00012400
 94             XLA1 = 'CALCULATE XN'                                      .00012500
 95             XLA2 = 'CALCULATE FN FROM XN'                              .00012600
 96             XLB = 'LOOP CONTROL'                                       .00012700
 97             XLXA = 'SELECT NEW XA'                                     .00012800
 98             XLXB  = 'SELECT NEW XB'                                    .00012900
 99             XLFA = 'SELECT NEW FA'                                     .00013000
100             XLFB = 'SELECT NEW FB'                                     .00013100
101             Y.STEP.A =  XLA1 DUPL(' ',25 - SIZE(XLA1)) 'STEP A  '      .00013200
102             Y.STEP.B =  XLA2 DUPL(' ',25 - SIZE(XLA2)) 'STEP B  '      .00013300
103             Y.STEP.C =  XLB  DUPL(' ',25 - SIZE(XLB))  'STEP C  '      .00013400
104             Y.STEP.D =  XLXA DUPL(' ',25 - SIZE(XLXA)) 'STEP D  '      .00013500
105             Y.STEP.E =  XLFA DUPL(' ',25 - SIZE(XLFA)) 'STEP E  '      .00013600
106             Y.STEP.F =  XLXB DUPL(' ',25 - SIZE(XLXB)) 'STEP F  '      .00013700
107             Y.STEP.G =  XLFB DUPL(' ',25 - SIZE(XLFB)) 'STEP G  '      .00013800
108             PS.ALL = 'AT THIS POINT IN THE ANALYSIS OF SUBROUTINE '    .00013900
109             PS.ALL.1 = ' THE PLAUSABILITY IS:  '                       .00014000
110             P.U. = 'THIS POSITION IS UNDEFINED'                        .00014100
111             STAR = DUPL('* *',3) '    '                                .00014200
112             OUTPUT = 'WELCOME TO THE ALGORITHM CHECKER'                .00014300
113            &TRIM = 0                                                   .
    **************  INITIALIZE                       ******************** .00014400
114 L0          PROB.SUCCESS = 100.0                                       .00014500
115             TIMEOUT()                                                  .00014600
116             RETURN =                                                   .00014700
117             FUNCTION.X =                                               .00014800
118             NPARM = 0                                                  .00014900
119             RESULT = 0                                                 .00015000
120             XAR = ; XBR =                                              .00015100
122             XA =                                                       .00015200
123             XB =                                                       .00015300
124             XN =                                                       .00015400
125             FA =                                                       .00015500
126             FB =                                                       .00015600
127             FN =                                                       .00015700
128             FUNCTION.T =                                               .00015800
129             ZLA = ARRAY(5)                                             .00015900
130             ZLB = ARRAY(5)                                             .00016000
131             ZRA = ARRAY(5)                                             .00016100
132             ZRB = ARRAY(5)                                             .00016200
133             ZNA = ARRAY(5)                                             .00016300
134             ZNB = ARRAY(5)                                             .00016400
135             PARM = ARRAY(10)                                           .00016500
136             I.STEP.A =                                                 .00016600
137             I.STEP.B =                                                 .00016700
138             I.STEP.C =                                                 .00016800
139             I.STEP.D =                                                 .00016900
140             I.STEP.E =                                                 .00017000
141             L.STEP.A1 = ; L.STEP.A2 =                                  .00017100
143             L.STEP.B =                                                 .00017200
```

```
144              L.STEP.C.XA = ; L.STEP.C.XB =                              .00017300
146              L.STEP.C.FA = ; L.STEP.C.XB =                              .00017400
148              FUNCTION =                                                 .00017500
149              ST = TABLE()                                               .00017600
150              STN = ARRAY(50)                                            .00017700
151              LDECK = ARRAY(50)                                          .00017800
152              UDECK = ARRAY(50)                                          .00017900
153              DECK = ARRAY(50)                                           .00018000
154              DELTA =                                                    .00018100
155              OUTPUT =                                                   .00018200
156              OUTPUT = DUPL(STAR,8)                                      .00018300
157              OUTPUT = DUPL('*START HERE*',8)                            .00018400
158              OUTPUT = 'THIS IS THE INPUT DECK'                          .00018500
159              J = 1                                                      .00018600
160              I = 0                                                      .00018700
161 L1           INPUT LEN(72) . CARD REM . SEQ.NUM      :F(END)            .00018800
162              OUTPUT = CARD DUPL(' ',72 - SIZE(CARD)) SEQ.NUM            .00018900
163              J = J + 1                                                  .00019000
164              CARD COMMENT                            :S(L1)             .00019100
    ************************************************************************ .00019200
    *************                                        ******************* .00019300
    *************  STATEMENT NUMBER ????                 ******************* .00019400
    *************                                        ******************* .00019500
    ************************************************************************ .00019600
165              CARD POS(0) DUPL(' ',5)                 :S(L2)             .00019700
166              CARD POS(0) LEN(6) . N =                                   .00019800
    *************  CLEAR BLANKS                                             .00019900
167 L4           N FIELD = STR                           :S(L4)             .00020000
168              INTEGER(N)                              :S(L9)             .00020100
169              N =                                                        .00020200
170 L9           ST<N> = I + 1                                              .00020300
171              STN<I + 1> = N                                             .00020400
172 L3           I = I + 1                                                  .00020500
173 L5           CARD FIELD = STR                        :S(L5)             .00020600
174              DECK<I> = CARD                                             .00020700
175              CARD POS(0) 'END' RPOS(0)               :F(L1)             .00020800
176              OUTPUT = DUPL('*',100)                                     .00020900
177              DECKOUT()                               :(L10)             .00021000
    *************                                                           .00021100
    *************  TEST FOR CONTINUATION                                    .00021200
    *************                                                           .00021300
178 L2           CARD POS(0) ' ' =                       :S(L3)             .00021400
    *************  BRANCH IF NO CONTINUATION                                .00021500
179 L7           CARD FIELD = STR                        :S(L7)             .00021600
    *************  CLEAR BLANKS                                             .00021700
180              DECK<I> = DECK<I> CARD.                 :(L1)              .00021800
```

```
    ************************************************************************* .00022000
    *************                                         ******************* .00022100
    *************  TO FIND AND ISOLATE THE LOOP STEP      ******************* .00022200
    *************                                         ******************* .00022300
    ************************************************************************* .00022400
    *************                                         ******************* .00022500
    *************                                         ******************* .00022600
    *************  TYPE 1                                 ******************* .00022700
    *************  N  --------                            ******************* .00022800
    *************     --------                            ******************* .00022900
    *************     --------                            ******************* .00023000
    *************     IF(?? .GT. DELTA) GO TO N           ******************* .00023100
    *************                                         ******************* .00023200
    ************* TYPE 2                                  ******************* .00023300
    *************                                         ******************* .00023400
    *************  M  --------                            ******************* .00023500
    *************     --------                            ******************* .00023600
    *************     --------                            ******************* .00023700
    *************     IF(??.LT.DELTA)GO TO N              ******************* .00023800
    *************     --------                            ******************* .00023900
    *************     GO TO M                             ******************* .00024000
    *************     --------                            ******************* .00024100
    *************     --------                            ******************* .00024200
    *************  N  --------                            ******************* .00024300
    *************                                         ******************* .00024400
    ************************************************************************* .00024500
    *************  IT IS ASSUMED THAT THE ALGORITHM WILL  ******************* .00024600
    ************* TERMINATE IN A FINITE NUMBER OF STEPS   ******************* .00024700
    ************* THEREFORE NO DO LOOP NEED EXIST IN THE  ******************* .00024800
    ************* "IDEAL" MECHANIZATION OF THIS ALGORITHM******************** .00024900
    ************************************************************************* .00025000
181 L10          TIMEOUT() ; PROBOUT()                                        .00025100
183              OUTPUT = STAR 'LIST VARIABLES PASSED FROM MAIN PROGRAM'      .00025200
184              ' - IF ANY'                                                  .00025250
185              II = 0                                                       .00025300
186              KK = 1                                                       .00025400
187 LOH          KK =  LT(KK,I) KK + 1                    :F(LOI)             .00025500
188              DECK<KK> POS(0) 'COMMON' (VAR ARB) . TEMP RPOS(0)            .00025600
    +                                                     :F(LOH)             .00025700
189              UDECK<KK> = 'COMMON STATEMENT'                               .00025800
190 LON          TEMP VAR . PARM<II + 1> (BREAK(',') | RPOS(0)) =             .00025900
    +                                                     :F(LOI)             .00026000
191              II = II + 1                              :(LON)              .00026100
192 LOI          DECK<1> POS(0) 'SUBROUTINE' SPAN(ALPHAMERIC) . NAME          .00026200
    +            '(' VAR . PARM<II + 1> BREAK(',)') ARB . TEMP RPOS(0)        .00026300
    +                                                     :F(LLO)             .00026400
193 PARM.1       II = II + 1                                                  .00026500
194              TEMP POS(0) ',' VAR . PARM<II + 1> BREAK(',)') =             .00026600
    +                                                     :S(PARM.1)          .00026700
195              TEMP POS(0) ')' RPOS(0)                  :F(LLO)             .00026800
196              OUTPUT = 'THE SUBROUTINE IS NAMED:  ' NAME                   .00026900
197              OUTPUT = 'THE MAIN PROGRAM PASSED THESE PARAMETERS:'         .00027000
198              JJ = 1                                                       .00027100
199              NPARM = II                                                   .00027200
200 PARM.2       OUTPUT = DUPL(' ',6 - SIZE(JJ)) JJ DUPL(' ',5) PARM<JJ>      .00027300
201              JJ = LT(JJ,II) JJ + 1                    :S(PARM.2)          .00027400
```

```
202             UDECK<I> = 'SUBROUTINE STATEMENT'    :(LOF)            .00027500
203 LLO         OUTPUT = 'THIS IS NOT A SUBROUTINE - ABORT'            .00027600
204             OUTPUT =                             :(LO)             .00027700
205 LOF         OUTPUT = STAR 'LEVEL ONE - IS TOP FLOW CORRECT - '     .00027800
    +           'FIND LOOP STEP'                                       .00027900
206             OUTPUT = 'ELIMINATE DO LOOPS, IF ANY, FROM THE '       .00028000
    +           'INPUT PROGRAM - REPLACE THEM WITH INFINITE LOOPS'     .00028100
207 LOFX        II = 1                                                 .00028200
208 LOB         II = LT(II,I) II + 1                 :F(LOA)           .00028300
209             DECK<II> DO.STATEMENT                :F(LOB)           .00028400
210             IDENT(STN<II>)                       :F(LOC)           .00028500
211             STN<II> = '$' II                                       .00028600
212 LOC         IY = STN<II>                                           .00028700
213             ST<IY> = II                                            .00028800
214 LOD         DECK<II> = 'CONTINUE'                                  .00028900
215             IX = ST<XSTN>                                          .00029000
216             I = I + 1                                              .00029100
217             II = I + 1                                             .00029200
218 LOG         II = GT(II,IX + 2) II - 1            :F(LOE)           .00029300
219             STN<II> = STN<II - 1>                                  .00029400
220             DECK<II> = DECK<II - 1>                                .00029500
221             ST<STN<II>> = DIFFER(STN<II>) ST<STN<II>> + 1          .00029600
    +                                                :(LOG)            .00029700
222 LOE         DECK<IX + 1> = 'GOTO' IY                               .00029800
223             STN<IX + 1> =                                          .00029900
224             OUTPUT = 'THE DO LOOP IS SUPERFLUOUS AND HAS BEEN REMOVED' .00030000
    +                                                :(LOFX)           .00030100
225 LOA         II = 0                                                 .00030200
226 L11         II = LT(II,I) II + 1                 :F(L14)           .00030300
227 L11Y        DECK<II> IFGOTO                      :S(L11X)          .00030400
228             DECK<II> IFRETURN                    :F(L11)           .00030500
229             DECK<I> = 'RETURN'                                     .00030600
230             STN<I> = '$' I                                         .00030700
231             ST<STN<I>> = I                                         .00030800
232             DECK<II> = 'IF(' LOGEXP ')GOTO' '$' I                  .00030900
233             I = I + 1                                              .00031000
234             DECK<I> = 'END'                      :(L11Y)           .00031100
235 L11X     TOTAL = 0                                                 .00031200
236 LX0      LOGEXP POS(0) '.NOT.(' LGXP ')' RPOS(0) :F(LX1)           .00031300
237          TOTAL = TOTAL + 1                    :(LX0)               .00031400
238 LX1      EQ(TOTAL,0)                          :S(LX2)              .00031500
239          EQ(TOTAL,-1)                         :S(LX3)              .00031600
240          TOTAL = TOTAL - 2                    :(LX1)               .00031700
241 LX3      OP LESS = 'GT'                       :S(LX2)              .00031800
242          OP = 'LT'                                                 .00031900
243 LX2      EXP1 ZERO                            :S(LX4)              .00032000
244             EXP2 ZERO                            :F(LX5)           .00032100
245             EXP2 POS(0) BAL . TEMP '-' BAL . EXP1 RPOS(0)          .00032200
    +                                                :F(LX5)           .00032300
246             EXP2 = TEMP                          :(LX5)            .00032400
247 LX4         EXP1 POS(0) BAL . TEMP '-' BAL . EXP2 RPOS(0)          .00032500
    *                                                :F(LX5)           .00032600
248             EXP1 = TEMP                                            .00032700
249 LX5         XST2 = XST                                             .00032800
    **************  XST2 IS THE DESTINATION OF THE      ******************** .00032900
    **************  IF - GO TO STATEMENT                ******************** .00033000
```

```
250               LT(ST<XST>,II)                          :F(L11A)          .00033100
251               IJK = ST<XST>                                             .00033200
252 L11B          GT(IJK,II)                              :S(L11C)          .00033300
253               LDECK<IJK> = 'LOOP'                                       .00033400
254               IJK = IJK + 1                           :(L11B)           .00033500
255 L11C          UDECK<II> = 'LOOP CONTROL'                                .00033600
256               L.STEP.B = II                                             .00033700
257               OUTPUT = 'SUCCESS - MATCH TO PATTERN TYPE ONE'            .00033800
    +                                                     :(L13)            .00033900
258 L11A          JJ = II                                                   .00034000
259 L16           JJ = LT(JJ,ST<XST2>)  JJ + 1            :F(L11)           .00034100
260               DECK<JJ> GCTCST                         :F(L16)           .00034200
261               LE(ST<XST>,II)                          :F(L16)           .00034300
262               LOGEXP = EXP2 OP EXP1                                     .00034400
263               LOGEXP LLEXP = EXP1 '.LT.' EXP2         :F(L17)           .00034500
264               LOGEXP LESS    = '.LT.'                 :F(L11)           .00034600
265 L17           DECK<II> = 'IF(' LOGEXP  ')GOTO' XST2                     .00034700
266               UDECK<II> = 'LOOP CONTROL'                                .00034800
267               L.STEP.B = II                                             .00034900
268               IJK = ST<XST> - 1                                         .00035000
269 L17C          IJK = LT(IJK,ST<XST2> - 1) IJK + 1      :F(L17B)          .00035100
270               LDECK<IJK> = 'LOOP'                     :(L17C)           .00035200
271 L17B          OUTPUT = 'SUCCESS - MATCH TO PATTERN TYPE TWO'            .00035300
    ********************************************************************** .00035400
    *************                                     ******************** .00035500
    *************  IF SUCCESS OUTPUT MODIFIED DECK    ******************** .00035600
    ************* IDENTIFY AND MARK OFF               ******************** .00035700
    *************   INITIALIZE AND TERMINATE STEPS    ******************** .00035800
    *************                                     ******************** .00035900
    ********************************************************************** .00036000
272 L13           M = 0                                                     .00036100
273 L13E          M = LT(M,I) M + 1                       :F(L13C)          .00036200
274               LDECK<M> 'LOOP'                         :S(L13A)          .00036300
275               LDECK<M> = 'INITIALIZE'                 :(L13E)           .00036400
276 L13A          M = LT(M,I) M + 1                       :F(L13C)          .00036500
277               LDECK<M> 'LOOP'                         :F(L13B)S(L13A)   .00036600
278 L13D          M = LT(M,I) M + 1                       :F(L13C)          .00036700
279 L13B          LDECK<M> = 'TERMINATE'                  :(L13D)           .00036800
280 L13C          DECKOUT()                                                 .00036900
281               PROBOUT()                               :(L20)            .00037000
282 L14           PROB.SUCCESS = PROB.SUCCESS * PROB.A                      .00037100
283               OUTPUT = 'FAILED TO FIND THE LOOP STEP -'                 .00037200
    +             ' HOPELESS CASE - TRY NEXT PROGRAM'                       .00037300
284               PROBOUT()                               :(L0)             .00037400
```

```
    *********************************************************************** .00037600
    **************                                      ******************* .00037700
    ************** MATCH TO FORM:IF(? .GT. DELTA)GOTO ?    ***************** .00037800
    ************** CONTINUE TO LOOP IF LOGICAL EXPRESSION  ***************** .00037900
    **************                  IS TRUE              ******************* .00038000
    **************                                      ******************* .00038100
    *********************************************************************** .00038200
285              OUTPUT = STAR 'ADDITIONAL HEURISTIC - FIND TERMINATION '    .00038300
    +            'VALUE MAGNITUDE NOW, NOT LATER'                            .00038400
286 L20          DELTA = EXP2                                                .00038500
287              OUTPUT = FZERO(EXP1) 'LEFT HAND EXPRESSION IS ZERO '        .00038600
    +            '- REWORK LOGICAL EXPRESSION'          :S(L37)              .00038700
288              OUTPUT = FZERO(DELTA) 'FIRST GUESS AT DELTA IS ZERO '       .00038800
    +            '- REWORK LOGICAL EXPRESSION'          :S(L36)              .00038900
289              REAL(DELTA)                            :S(L31)              .00039000
290              DELTA POS(0) INT RPOS(0)               :S(L31)              .00039100
291 L20A         VARIABLE(DELTA)                        :F(L21)              .00039200
    **************                                      ******************* .00039300
    ************** IF VARIABLE - SCAN INPUT FOR ACTUAL   ******************* .00039400
    **************         VALUE                        ******************* .00039500
292 L23B         SCAN.PARM(DELTA)                       :F(L23BX)            .00039600
293              OUTPUT = 'DELTA WAS PASSED TO SUBROUTINE FROM CALLING '     .00039700
    +            'PROGRAM'                              :(L31)               .00039800
294 L23BX        MN = 0                                                      .00039900
295 L23          MN = LT(MN,I) MN + 1                   :F(L22)              .00040000
296              CARD = DECK<MN>                                             .00040100
297              CARD POS(0) DELTA '=' =                :F(L23)              .00040200
298              UDECK<MN> = 'DELTA DEFINED'                                 .00040300
299              VARIABLE(CARD)                         :S(L23A)             .00040400
300              CARD POS(0) NUMBER . DELTA RPOS(0)     :S(L31)F(L21)        .00040500
301 L22          SCAN.PARM(DELTA)                       :F(L22A)             .00040600
302              OUTPUT =  DELTA ' WAS PASSED TO SUBROUTINE FROM '           .00040700
    +            'CALLING PROGRAM'                      :(L31A)              .00040800
303 L22A         OUTPUT = 'FAILURE TO FIND VARIABLE NAMED "' DELTA           .00040900
    +            '" - ASSUME DELTA UNDEFINED'                                .00041000
304              PROB.SUCCESS = PROB.SUCCESS * PROB.B   :(L40)               .00041100
305 L21          DELTA POS(0) EXP RPOS(0)               :F(L21A)             .00041200
306              OUTPUT = 'CANNOT ISOLATE DELTA FROM EXPRESSION "'           .00041300
    +            DELTA '"'                              :(L21B)              .00041400
307 L21A         OUTPUT = 'CANNOT FIND VALUE OF DELTA - ASSUME DELTA '       .00041500
    +            'UNDEFINED'                                                 .00041600
308 L21B         PROB.SUCCESS = PROB.SUCCESS * PROB.C   :(L31A)              .00041700
309 L31          OUTPUT = 'DELTA IS OF THE MAGNITUDE  ' DELTA                .00041800
    +                                                   :(L31A)              .00041900
    **************  ASSUME EXP1 IS OF THE FORM "EXP - DELTA'                 .00042000
310 L36          EXP1 '-' (VAR | NUMBER) . DELTA RPOS(0) =                   .00042100
    +                                                   :F(L22Z)             .00042200
311              EXP2 = DELTA                                                .00042300
312 L23C         LOGEXP = EXP1 OP EXP2                                       .00042400
313              DECK<II> = 'IF(' LOGEXP ')GOTO' XST2   :(L23)               .00042500
    **************  ASSUME EXP2 IS OF THE FORM "DELTA - EXP                  .00042600
314 L37          EXP2 POS(0) (VAR | NUMBER) . DELTA '-'  =                   .00042700
    +                                                   :F(L22Z)             .00042800
315              EXP1 = EXP2                                                 .00042900
316              EXP2 = DELTA                           :(L23C)              .00043000
317 L23A         DELTA = CARD                           :(L23B)              .00043100
```

```
318 L22Z          OUTPUT = 'LOGICAL EXPRESSION DOES NOT MATCH AN ACCEPTABLE ' .00043200
    +             'PATTERN'                                                   .00043300
319               OUTPUT = 'MAGNITUDE AND NAME OF DELTA UNDEFINED'            .00043400
320               PROB.SUCCESS = PROB.SUCCESS * PROB.D     :(L40)             .00043500
321 L31A          MN = 0                                                      .00043600
322               TV = EXP1                                                   .00043700
323 L31Z          TV ABSF = YCT                            :S(L32)            .00043800
324               VARIABLE(TV)                             :F(L33)            .00043900
325 L34           MN = LT(MN,I) MN + 1                     :F(L33A)           .00044000
326               DECK<MN> POS(0) TV '='                   :F(L34)            .00044100
327               TV = DECK<MN>                                               .00044200
328               TV ARB '=' =                             :(L31Z)            .00044300
329 L32           OUTPUT =    'TERMINATION IS RELATIVE TO ABSOLUTE '          .00044400
    +             'VALUE OF  ' TV                          :(L40)             .00044500
330 L33           TV PVV2                                  :F(L33A)           .00044600
331               OUTPUT =   'TERMINATION IS RELATIVE TO THIS POSITIVE'       .00044700
    +             ' FUNCTION  ' TV                         :(L40)             .00044800
332 L33A          OUTPUT = 'THE TERMINATION CONTROL "' TV '" IS NOT A '       .00044900
    +             'POSITIVE VALUED FUNCTION'                                  .00045000
333               PROB.SUCCESS = PROB.SUCCESS * PROB.E     :(L40)             .00045100
334 L40           PROBOUT()                                                   .00045200
```

```
    **************  INITIALIZE STEP                          ******************** .00045400
    **************************************************************************** .00045500
    **************  LEVEL TWO                                ******************** .00045600
    **************************************************************************** .00045700
335               OUTPUT = DUPL('*',100)                                         .00045800
336               TIMEOUT()                                                      .00045900
337               OUTPUT = STAR 'LEVEL TWO - INITIALIZE STEP'                    .00046000
338               OUTPUT = STAR 'FIND FUNCTION DEFINITION'                       .00046100
339               DECKOUT()                                                      .00046200
    **************************************************************************** .00046300
    **************                                           ******************** .00046400
    ************** TEST ONE - FIND FUNCTION DEFINITION      ******************** .00046500
    **************                                           ******************** .00046600
    **************************************************************************** .00046700
340               OUTPUT = STAR 'TEST ONE - FIND FUNCTION DEFINITION'            .00046800
    +             ' - ASSUME FUNCTION DEFINED BY FUNCTION STATEMENT'             .00046900
341               II = 0                                                         .00047000
342 L71           II = II + 1                                                    .00047100
343               IDENT(LDECK<II>,'INITIALIZE')             :F(L70)              .00047200
344               DECK<II> POS(0) VAR . FUNCTION '(' VAR . X ')='                .00047300
    +                                                       :F(L71)              .00047400
345               FUNCTION.T = DECK<II>                                          .00047500
346               FUNCTION.T POS(0) FUNCTION '(' X ')=' =                        .00047600
347               FUNCTION.T ARB . TEMP X =                 :F(L17B)             .00047700
348               TEMP = TEMP 'X'                                                .00047800
349 L71A          FUNCTION.T ARB . TEMP1 X =                :F(L71C)             .00047900
350               TEMP = TEMP TEMP1 'X'                     :(L71A)              .00048000
351 L71C          FUNCTION.T = TEMP FUNCTION.T                                   .00048100
352               UDECK<II> = 'FUNCTION DEFINED'                                 .00048200
353               I.STEP.A = II                                                  .00048300
354               VAROUT()                                  :(L72)               .00048400
355 L70           OUTPUT = 'THE FUNCTION WE WANT CANT BE FOUND'                  .00048500
    +             ', EITHER:'                                                    .00048600
356               OUTPUT = DUPL(' ',10) '1.  THE FUNCTION IS UNDEFINED'          .00048700
357               OUTPUT = DUPL(' ',10) '2.  THE FUNCTION IS EXTERNALLY '        .00048800
    +             'DEFINED'                                                      .00048900
358               OUTPUT = DUPL(' ',10) '3.  THE FUNCTION IS WRITTEN OUT '       .00049000
    +             'EACH TIME'                                                    .00049100
359               PROB.SUCCESS = PROB.SUCCESS * PROB.F      :(L72)               .00049200
360 L71B          FUNCTION = ; FUNCTION.T =                 :(L71)               .00049300
```

```
    ************************************************************************** .00049500
    *************                                         ******************** .00049600
    ************* TEST TWO                                ******************** .00049700
    ************* FIND INITIALIZATION OF XA AND XB        ******************** .00049800
    *************                                         ******************** .00049900
    ************************************************************************** .00050000
362 L72          OUTPUT = STAR 'TEST TWO - FIND INITIALIZATION OF '           .00050100
    +            'XA AND XB'                                                   .00050200
363              II = 0                                                        .00050300
364 L74          II = II + 1                                                   .00050400
365              DIFFER(LDECK<II>,'INITIALIZE')           :S(L75)              .00050500
366              IDENT(UDECK<II>)                         :F(L74)              .00050600
367              DECK<II> SIMPLE.AS.N2                    :F(L74)              .00050700
368              XA = X                                                        .00050800
369              XAR = XR                                                      .00050900
370              UDECK<II> = 'INITIALIZE XA'                                   .00051000
371              I.STEP.B = II                                                 .00051100
372 L76          II = IDENT(LDECK<II>,'INITIALIZE') II + 1                     .00051200
    +                                                     :F(L77)              .00051300
373              IDENT(UDECK<II>)                         :F(L76)              .00051400
374              DECK<II> SIMPLE.AS.N2                    :F(L76)              .00051500
375              XB = X                                                        .00051600
376              XBR = XR                                                      .00051700
377              UDECK<II> = 'INITIALIZE XB'                                   .00051800
378              I.STEP.C = II                            :(L80)               .00051900
379 L75          OUTPUT = 'NEITHER XA OR XB WAS INITIILIZED BY SUBROUTINE'     .00052000
380              PROB.SUCCESS = PROB.SUCCESS * PROB.G                          .00052100
    +            :(L80)                                                        .00052200
381 L77          OUTPUT = 'ONLY XA WAS INITIALIZED'                            .00052300
382              PROB.SUCCESS = PROB.SUCCESS * PROB.H     :(L80)               .00052400
```

```
    ****************************************************************** .00052600
    *************                                  ******************* .00052700
    ************* TEST  THREE                      ******************* .00052800
    *************  FIND CALCULATION OF FA AND FB   ******************* .00052900
    *************                                  ******************* .00053000
    ****************************************************************** .00053100
383 L80           PROBOUT() ; TIMEOUT()                                .00053200
385               OUTPUT = STAR 'FIND CALCULATION OF FA AND FB'        .00053300
386               TF = IDENT(FUNCTION) VAR . FUNCTION      :S(L80A)    .00053400
387               TF = FUNCTION                                        .00053500
388 L80A          IDENT(XA XB)                             :F(L80C)    .00053600
389               TX1 = VAR . XA                                       .00053700
390               TX2 = VAR . XB                           :(L80E)     .00053800
391 L80C          IDENT(XB)                                :F(L80D)    .00053900
392               TX1 = XA | XAR                                       .00054000
393               TX2 = VAR . XB                           :(L80E)     .00054100
394 L80D          TX1 = XA | XAR                                       .00054200
395               TX2 = XB | XBR                                       .00054300
396 L80E          II = 0                                               .00054400
397 L81           II =  II + 1                                         .00054500
398               IDENT(LDECK<II>,'INITIALIZE')            :F(L84)     .00054600
399               IDENT(UDECK<II>)                         :F(L81)     .00054700
400               DECK<II> POS(0) VAR . FA '=' TF '(' TX1 ')' RPOS(0)  .00054800
    +                                                      :F(L81)     .00054900
401               UDECK<II> = 'CALCULATE FA'                           .00055000
402               I.STEP.D = II                                        .00055100
403               RESULT = 1                                           .00055200
404 L84           II = 0                                               .00055300
405 L85           II = II + 1                                          .00055400
406               IDENT(LDECK<II>,'INITIALIZE')            :F(L86)     .00055500
407               IDENT(UDECK<II>)                         :F(L85)     .00055600
408               DECK<II> POS(0) VAR . FB '=' TF '(' TX2 ')' RPOS(0)  .00055700
    +                                                      :F(L85)     .00055800
409               I.STEP.E = II                                        .00055900
410               UDECK<II> = 'CALCULATE FB'                           .00056000
411               RESULT = 1 + RESULT                                  .00056100
412 L86           IDENT(RESULT)                            :F(L86A)    .00056200
413               OUTPUT = 'THERE ARE NO CALCULATIONS OF EITHER FA '   .00056300
    +             'OR FB USING AN ALREADY DEFINED FUNCTION STATEMENT'  .00056400
414 L86A          IDENT(RESULT FUNCTION)                   :F(L90)     .00056500
415               II = 0                                               .00056600
416 L87A          II = II + 1                                          .00056700
417               IDENT(LDECK<II>,'INITIALIZE')            :F(L87B)    .00056800
418            TEMP = DECK<II>                                         .00056900
419               TEMP POS(0) VAR . TEMP3 '=' =            :F(L87A)    .00057000
420               TEMP ARB . TEMP1 TX1 =                   :F(L87A)    .00057100
421               TEMP1 = TEMP1 'X'                                    .00057200
422 L87F          TEMP ARB . TEMP2 TX1 =                   :F(L87M)    .00057300
423               TEMP1 = TEMP1 TEMP2 'X'                  :(L87F)     .00057400
424 L87M          FUNCTION.T = TEMP1 TEMP                              .00057500
425               FA = TEMP3                                           .00057600
426               UDECK<II> = 'CALCULATE FA'                           .00057700
427               OUTPUT = 'FA IS CALCULATED IN - ' DECK<II>           .00057800
428               RESULT = 1                                           .00057900
429 L87B          II = 0                                               .00058000
430 L87C          II = II + 1                                          .00058100
```

```
                IDENT(LDECK<II>,'INITIALIZE')             :F(L90)             .00058200
             TEMP = DECK<II>                                                  .00058300
                TEMP POS(0) VAR . TEMP3 '=' =             :F(L87C)            .00058400
                TEMP ARB . TEMP1 TX2 =                    :F(L87C)            .00058500
                TEMP2 = TEMP1 'X'                                             .00058600
     L87G       TEMP1 ARB . TEMP2 TX2 =                   :F(L87N)            .00058700
                TEMP1 = TEMP1 TEMP2 'X'                   :(L87G)             .00058800
     L87N       TEMP = TEMP1 TEMP                                             .00058900
                RESULT = RESULT + 1                                           .00059000
                UDECK<II> = 'CALCULATE FB'                                    .00059100
                CONFLICT(FUNCTION.T,TEMP,'THE EXPRESSION WHICH DEFINES '      .00059200
      +         'THE FUNCTION')                                               .00059300
                PROB.SUCCESS = EQ(1,RESULT) PROB.SUCCESS * PROB.J             .00059400
                TIMEOUT() : PROBOUT()                                         .00059500
```

```
    ******************************************************************** .00059700
    *************                                      ****************** .00059800
    ************* STEP FOLR                            ****************** .00059900
    ************* CHECK PRECICENCE IN INITIALIZE STEP  ****************** .00060000
    *************                                      ****************** .00060100
    ******************************************************************** .00060200
445 L90          PROB.SUCCESS = DIFFFF(I.STEP.A) DIFFER(I.STEP.B)        .00060300
    +                           GE(I.STFP.A,I.STEP.C) PROB.SUCCESS *    .00060400
    +            PROB.K                                                  .00060500
446              PROB.SUCCESS = DIFFER(I.STEP.A) DIFFER(I.STFP.B)        .00060600
    +                           GE(I.STEP.A,I.STEP.B) PROB.SUCCESS *    .00060700
    +            PROB.K                                                  .00060800
447              PROB.SUCCESS = DIFFER(I.STEP.A) DIFFER(I.STEP.D)        .00060900
    +                           GE(I.STEP.A,I.STEP.D) PROB.SUCCESS *    .00061000
    +            PROB.K                                                  .00061100
448              PROB.SUCCESS = DIFFEF(I.STEP.A) DIFFER(I.STEP.E)        .00061200
    +                           GE(I.STFP.A,I.STEP.F) PROB.SUCCESS *    .00061300
    +            PROB.K                                                  .00061400
449              PROB.SUCCESS = DIFFER(I.STEP.B) DIFFFR(I.STEP.D)        .00061500
    +                           GE(I.STEP.B,I.STEP.D) PROB.SUCCESS *    .00061600
    +            PROB.K                                                  .00061700
450              PROB.SUCCESS = DIFFEF(I.STEP.C) DIFFER(I.STEP.E)        .00061800
    +                           CE(I.STEP.C,I.STEP.E) PROB.SUCCESS *    .00061900
    +            PROB.K                                                  .00062000
451              OUTPUT = 'THE EXPECTED PPECIDENCE IS'                   .00062100
452              OUTPUT =                                                .00062200
453              OUTPUT = '     A    B    C    D    E'                   .00062300
454              OUTPUT = 'A    0'                                       .00062400
455              OUTPUT = 'B    1    0'                                  .00062500
456              OLTPUT = 'C    1    0    0'                             .00062600
457              OUTPUT = 'D    1    1    0    0'                        .00062700
458              OUTPUT = 'E    1    0    1    0    0'                   .00062800
459              OUTPUT =                                                .00062900
460              OUTPUT = 'THE ACTUAL PRECIDENCE IS:'                    .00063000
461              FILL = ICENT(I.STEP.A) P.U.                             .00063100
462              OLTPUT = CUPL(' ',10) X.STEP.A    I.STEP.A              .00063200
    +            FILL                                                    .00063300
463              FILL =                                                  .00063400
464              FILL = ICENT(I.STEP.B) P.U.                             .00063500
465              OUTPUT = CUPL(' ',10) X.STEP.B    I.STEP.B              .00063600
    +            FILL                                                    .00063700
466              FILL =                                                  .00063800
467              FILL = ICENT(I.STEP.C) P.U.                             .00063900
468              OUTPUT = DUPL(' ',10) X.STEP.C    I.STFP.C              .00064000
    +            FILL                                                    .00064100
469              FILL =                                                  .00064200
470              FILL =                                                  .00064300
471              FILL = ICENT(I.STEP.D) P.U.                             .00064400
472              OLTPUT = CUPL(' ',10) X.STFP.D    I.STEP.D              .00064500
    +            FILL                                                    .00064600
473              FILL =                                                  .00064700
474              FILL = ICENT(I.STEP.E) P.U.                             .00064800
475              OUTPUT = DUPL(' ',10) X.STEP.E    I.STEP.E              .00064900
    +            FILL                                                    .00065000
476              FILL =                                                  .00065100
477              TIMEOUT() : PROBCUT()                                   .00065200
```

```
    *********************************************************************** .00065400
    ************** LEVEL TWO                              ******************** .00065500
    ************** LOOP STEP                              @******************* .00065600
    *********************************************************************** .00065700
479               OUTPUT = DUPL('*',100)                                    .00065800
480               OUTPUT = TIME()                                           .00065900
481               OUTPUT = STAR 'TEST ONE - FIND CALCULATION OF XN, FIND'   .00066000
   +              ' CALCULATION OF FN FROM XN'                              .00066100
482               ST.PARM = ARRAY(10) ; ST.PARM.F = ARRAY(10)               .00066200
484               TX1 = VAR . XA . TX1                                      .00066300
485               TX2 = VAR . XB . TX2                                      .00066400
486               TX3 = VAR . FA . TX3                                      .00066500
487               TX4 = VAR . XB . TX4                                      .00066600
488               TF = IDENT(FUNCTION) VAR . FUNCTION     :S(P10D)          .00066700
489               TF = FUNCTION                                             .00066800
490 P10D          II = 0                                                    .00066900
491 P10           II = II + 1                                               .00067000
492               IDENT(LDECK<II>,'LOOP')                  :F(P10)          .00067100
493 P12           DECK<II> POS(0) VAR . TEMPO '=' ARB . TEMP RPOS(0)        .00067200
   +                                                       :F(P12A)         .00067300
494               IDENT(UDECK<II>)                         :F(P12A)         .00067400
495               TEMP1 = TEMP                                              .00067500
496               JJ = 0                                                    .00067600
497 P10Z          TEMP1 POS(0) (ARB T.OP | NULL) VAR . ST.PARM<JJ + 1>      .00067700
   +              (BREAK('/-+*)') | RPOS(0)) =             :F(P10W)         .00067800
498               JJ = JJ + 1                                               .00067900
499               ST.PARM.F<JJ> = JJ                       :(P10Z)          .00068000
500 P10W          IDENT(XA)                                :S(P10Y)         .00068100
501               SCAN.VAR(XA)                             :F(P12A)         .00068200
502 P10U          SCAN.VAR(XA)                             :S(P10U)         .00068300
503 P10Y          IDENT(XB)                                :S(P10V)         .00068400
504               SCAN.VAR(XB)                             :F(P12A)         .00068500
505 P10T          SCAN.VAR(XB)                             :S(P10T)         .00068600
506 P10V          IDENT(FA)                                :S(P10S)         .00068700
507               SCAN.VAR(FA)                             :F(P12A)         .00068800
508 P10R          SCAN.VAR(FA)                             :S(P10R)         .00068900
509 P10S          IDENT(FB)                                :S(P10Q)         .00069000
510               SCAN.VAR(FB)                             :F(P12A)         .00069100
511 P10N          SCAN.VAR(FB)                             :S(P10N)         .00069200
512 P10Q          IDENT(XA)                                :F(P10M)         .00069300
513               SCAN.VAR(TX1)                            :F(P12A)         .00069400
514 P10L          SCAN.VAR(TX1)                            :S(P10L)         .00069500
515 P10M          IDENT(XB)                                :F(P10K)         .00069600
516               SCAN.VAR(TX2)                            :F(P12A)         .00069700
517 P10H          SCAN.VAR(TX2)                            :S(P10H)         .00069800
518 P10K          IDENT(FA)                                :F(P10J)         .00069900
519               SCAN.VAR(TX3)                            :F(P12A)         .00070000
520 P10G          SCAN.VAR(TX3)                            :S(P10G)         .00070100
521 P10J          IDENT(FB)                                :F(P11)          .00070200
522               SCAN.VAR(TX4)                            :F(P12A)         .00070300
523 P10F          SCAN.VAR(TX4)                            :S(P10F)F(P11)   .00070400
524 P12A          II = II + 1                                               .00070500
525               IDENT(LDECK<II>,'LOOP')                  :S(P12)F(P13)    .00070600
526 P11           TEMP POS(0) EXP . OUTPUT RPOS(0)         :F(P12A)         .00070700
527               XN = TEMPO                                                .00070800
528               REPLACE(TEMP,XA,'XA')                                     .00070900
```

```
529              REPLACE(TEMP,XB,'XB')                                          .00071000
530              REPLACE(TEMP,FA,'FA')                                          .00071100
531              REPLACE(TEMP,FB,'FB')                                          .00071200
532              FUNCTION.X = TEMP                                              .00071300
533              L.STEP.A1 = II                                                 .00071400
534              UDECK<II> = 'CALCULATE XN'                                     .00071500
535              OUTPUT = 'XN IS CALCULATED BY - ' TEMP                         .00071600
536 P18          II = II + 1                                                    .00071700
537              IDENT(LDECK<II>,'LOOP') IDENT(UDECK<II>) :F(P16)               .00071800
538              DECK<II> POS(0) VAR . FN '=' TF '(' XN ')' RPOS(0)             .00071900
    +                                                   :F(P18)                 .00072000
539 P17          UDECK<II> = 'CALCULATE FN'                                     .00072100
540              L.STEP.A2 = II                         :(P20)                  .00072200
541 P13          PROB.SUCCESS = PROB.SUCCESS * PROB.M                           .00072300
542              OUTPUT = 'FAILED TO FIND CALCULATION OF EITHER FN OR XN'       .00072400
    +            :(P20A)                                                        .00072500
543 P16          PROB.SUCCESS = PROB.SUCCESS * PROB.N                           .00072600
544              OUTPUT = 'FAILED TO FIND CALCULATION OF FN'                    .00072700
    +                                                   :(P20A)                 .00072800
545 P20          DECKOUT() : VAROUT()                                           .00072900
547 P20A         PROBOUT()                                                      .00073000
```

```
    ***************************************************************************** .00073200
    ************** FIND REPLACEMENT OF OLD XA OR XB          ******************** .00073300
    **************                                          ******************** .00073400
    **************       IF ( LOGEXP ) GO TO N              ******************** .00073500
    **************       X? = XN                            ******************** .00073600
    **************       F? = FN                            ******************** .00073700
    **************       GO TO M                            ******************** .00073800
    ************** N     X? = XN                            ******************** .00073900
    **************       F? = FN                            ******************** .00074000
    ************** M     --------                           ******************** .00074100
    **************                                          ******************** .00074200
    **************                                          ******************** .00074300
    ***************************************************************************** .00074400
548               OUTPUT = STAR 'FIND SELECTION OF NEW X AND ASSOCIATED F'      .00074500
549               II = 0                                                        .00074600
550 L41           II = II + 1                                                   .00074700
551               DIFFER(LDECK<II>,'LOOP')                 :S(L41)              .00074800
552 L42B          IDENT(UDECK<II>)                         :F(L42C)             .00074900
553 L42           DECK<II> IFGOTO                          :S(L41A)             .00075000
554 L42C          II = II + 1                                                   .00075100
555               DIFFER(LDECK<II>,'LOOP')                 :S(N90)F(L42B)       .00075200
556 L41A          XST2 = XST                                                    .00075300
557               BA = II + 1                                                   .00075400
558 L43           II = II + 1                                                   .00075500
559               DIFFER(LDECK<II>,'LOOP')                 :S(N90)              .00075600
560               DIFFER(UDECK<II>)                        :S(L43)              .00075700
561               DECK<II> GOTOST                          :F(L43)              .00075900
562               IDENT(ST<XST2>,II + 1)                   :F(L43)              .00075900
563               EA = II - 1                                                   .00076000
564               II = II + 1                                                   .00076100
565               BB = II                                                       .00076200
566               OUTPUT = 'WE HAVE PERFORMED A SUCCESSFUL PATTERN MATCH'       .00076300
    +             ' TO SELECTION PATTERN'                                       .00076400
567               UDECK<BA - 1> = 'SELECTION CONTROL'                           .00076500
568 L46           JJ = 0                                                        .00076600
569               II = BA - 1                                                   .00076700
570               OUTPUT = DUPL(' ',10) 'SECTION ONE'                           .00076900
571 L47           II = LT(II,EA) II + 1                    :F(L48A)             .00076900
572               IDENT(UDECK<II>)                         :F(L47)              .00077000
573               DECK<II> SIMPLEAS2                       :F(L47)              .00077100
574               OUTPUT = DUPL(' ',15) ZL ' = ' ZR                             .00077200
    +             DUPL(' ',15 - SIZE(ZL ZR))   II                               .00077300
575               JJ = JJ + 1                                                   .00077400
576               ZRA<JJ> = ZR                                                  .00077500
577               ZLA<JJ> = ZL                                                  .00077600
578               ZNA<JJ> = II                             :(L47)               .00077700
579 L48A          KK = 0                                                        .00077800
580               II = BB - 1                                                   .00077900
581               OUTPUT = DUPL(' ',10) 'SECTION TWO'                           .00078000
582 L48           II = IDENT(LDECK<II>,'LOOP') II + 1      :F(L49)              .00078100
583               DECK<II> GOTOST                          :S(L49)              .00078200
584               IDENT(UDECK<II>)                         :F(L48)              .00078300
585               DECK<II> SIMPLEAS2                       :F(L48)              .00078400
586               OUTPUT = DUPL(' ',15) ZL ' = ' ZR                             .00078500
    +             DUPL(' ',15 - SIZE(ZL ZR))   II                               .00078600
587               KK = KK + 1                                                   .00078700
```

```
588          ZRB<KK> = ZR                                                   .00078800
589          ZLB<KK> = ZL                                                   .00078900
590          ZNB<KK> = II                                 :(L48)            .00079000
591 L49      TX1 = IDENT(XA) VAR . XA                     :S(L49A)          .00079100
592          TX1 = XA                                                       .00079200
593 L49A     TX2 = IDENT(XB) VAR . XB                     :S(L49B)          .00079300
594          TX2 = XB                                                       .00079400
595 L49B     TF1 = IDENT(FA) VAR . FA                     :S(L49C)          .00079500
596          TF1 = FA                                                       .00079600
597 L49C     TF2 = IDENT(FB) VAR . FB                     :S(L49D)          .00079700
598          TF2 = FB                                                       .00079800
599 L49D     SCAN(ZLA,TX1,ZNA)                            :S(N2)            .00079900
600          SCAN(ZLA,TX2,ZNA)                            :S(N4)            .00080000
601          SCAN(ZLA,TF1,ZNA)                            :S(N6)            .00080100
602          SCAN(ZLA,TF2,ZNA)                            :S(N8)F(N9)       .00080200
603 N2       UDECK<XPCS>     =  'RECALC XA FROM XN'                         .00080300
604          CONFLICT(XN,ZRA<POS>,'XN')                                     .00080400
605          OUTPUT = 'SUCCESS - MATCH TO TX1'                              .00080500
606          L.STEP.C.XA = XPCS                                             .00080600
607          SCAN(ZLA,TF1,ZNA)                            :F(N19)           .00080700
608          L.STEP.C.FA = XPOS                                             .00080800
609          CONFLICT(FN,ZRA<POS>,'FN')                                     .00080900
610          UDECK<XPCS>     = 'RECALC FA FROM FN'    :(N50)                .00081000
611 N4       UDECK<XPCS>     = 'RECALC XB FROM XN'                          .00081100
612          CONFLICT(XN,ZRA<POS>,'XN')                                     .00081200
613          OUTPUT = 'SUCCESS - MATCH TO TX2'                              .00081300
614          L.STEP.C.XB = XPCS                                             .00081400
615          SCAN(ZLA,TF2,ZNA)                            :F(N15B)          .00081500
616          L.STEP.C.FB = XPCS                                             .00081600
617          UDECK<XPCS>     = 'RECALC FB FROM FN'                          .00081700
618          CONFLICT(FN,ZRA<POS>,'FN')                   :(N51)            .00081800
619 N15B     OUTPUT = 'FB NOT UPDATED '                                     .00081900
620          PROB.SUCCESS = PROB.SUCCESS * PROB.L         :(N51)            .00082000
621 N19      OUTPUT = 'FA NOT UPDATED'                                      .00082100
622          PROB.SUCCESS = PROB.SUCCESS * PROB.L         :(N50)            .00082200
623 N6       UDECK<XPCS>     = 'RECALC FA FROM FN'                          .00082300
624          CONFLICT(FN,ZRA<POS>,'FN')                                     .00082400
625          OUTPUT = 'SUCCESS - MATCH TO TF1'                              .00082500
626          L.STEP.C.FA = XPOS                                             .00082600
627          OUTPUT = 'XA NOT RECALC HERE'                                  .00082700
628          PROB.SUCCESS = PROB.SUCCESS * PROB.M         :(N50)            .00082800
629 N8       UDECK<XPCS>     = 'RECALC FB FROM FN'                          .00082900
630          OUTPUT = 'SUCCESS - MATCH TO TF2'                              .00083000
631          L.STEP.C.FB = XPCS                                             .00083100
632          CONFLICT(FN,ZRA<POS>,'FN')                                     .00083200
633 N25      OUTPUT = 'XB NOT RECALC HERE'                                  .00083300
634          PROB.SUCCESS = PROB.SUCCESS * PROB.M         :(N51)            .00083400
635 N50      CONFLICT(XN,ZRB<POS>,'XN')                                     .00083500
636 N54      UDECK<XPCS>     = 'RECALC XB FROM XN'                          .00083600
637          L.STEP.C.XB = XPCS                                             .00083700
638 N60      CONFLICT(FN,ZRB<POS>,'FN')                                     .00083800
639 N58      UDECK<ZNB<PCS>> = 'RECALC FB FROM FN'                          .00083900
640          L.STEP.C.FB = XPCS                           :(L90A)           .00084000
641 N51      SCAN(ZLB,TX1,ZNB)                            :F(N52X)          .00084100
642 N54X     UDECK<XPCS>     = 'RECALC XA FROM XN'                          .00084200
643          L.STEP.C.XA = XPCS                                             .00084300
```

```
644 N60X      SCAN(7LB,TF1,7NB)                           :F(N55X)             .00084400
645 N58X      UDECK<XPOS>      = 'RECALC FA FROM FN'                           .00084500
646           L.STEP.C.FA = XPOS                          :(L90A)              .00084600
647 N52       OUTPUT = 'XB NOT RECALC HERE'                                    .00084700
648           PROB.SUCCESS = PROB.SUCCESS * PROB.L        :(N60)               .00084800
649 N55       OUTPUT = 'FB NOT RECALC HERE'                                    .00084900
650           PROB.SUCCESS = PROB.SUCCESS * PROB.L        :(L90A)              .00085000
651 N52X      OUTPUT = 'XA NOT RECALC HERE'                                    .00085100
652           PROB.SUCCESS = PROB.SUCCESS * PROB.L        :(N60X)              .00085200
653 N55X      OUTPUT = 'FA NOT RECALC HERE'                                    .00085300
654           PROB.SUCCESS = PROB.SUCCESS * PROB.L        :(L90A)              .00085400
655 L90A      DECKOUT() ; VARCUT()                                             .00085500
657           PROBOUT()                                                        .00085600
658           TEMP = DIFFER(L.STEP.C.XA) L.STEP.C.XA  :S(LX10)                 .00085700
659           TEMP = DIFFER(L.STEP.C.XB) L.STEP.C.XB  :S(LX12)                 .00085800
660           TEMP = DIFFER(L.STEP.C.FA) L.STEP.C.FA  :S(LX13)F(LX11)          .00085900
661 LX10      TEMP = DIFFER(L.STEP.C.XB) GT(TEMP,L.STEP.C.XB) L.STEP.C.XB .00086000
662 LX12      TEMP = DIFFER(L.STEP.C.FA) GT(TEMP,L.STEP.C.FA) L.STEP.C.FA .00086100
663 LX13      TEMP = DIFFER(L.STEP.C.FB) GT(TEMP,L.STEP.C.FB) L.STEP.C.FB .00086200
664           PROB.SUCCESS = DIFFER(L.STEP.A1) GT(L.STEP.A1,TEMP)              .00086300
    +         PROB.SUCCESS * PROB.U                                            .00086400
665           PROB.SUCCESS = DIFFER(L.STEP.A2) GT(L.STEP.A2,TEMP)              .00086500
    +         PROB.SUCCESS * PROB.U                                            .00086600
666           PROB.SUCCESS = DIFFER(L.STEP.B) GT(L.STEP.B,TEMP)                .00086700
    +         PROB.SUCCESS * PROB.U                                            .00086800
667 LX11      PROB.SUCCESS = DIFFER(L.STEP.A1) DIFFER(L.STEP.A2)               .00086900
    +         GT(L.STEP.A1,L.STEP.A2) PROB.SUCCESS * PROB.U                    .00087000
668           PROB.SUCCESS = DIFFER(L.STEP.A1) DIFFER(L.STEP.B)                .00087100
    +         GT(L.STEP.A1,L.STEP.B) PROB.SUCCESS * PROB.U                     .00087200
669           PROB.SUCCESS = DIFFER(L.STEP.A2) DIFFER(L.STEP.B)                .00087300
    +         GT(L.STEP.A2,L.STEP.B) PROB.SUCCESS * PROB.U                     .00087400
670           OUTPUT = 'THE EXPECTED LOOP PRECEDENCE IS:'                      .00087500
671           OUTPUT =                                                         .00087600
672           OUTPUT = Y.STEP.A '1'                                            .00087700
673           OUTPUT = Y.STEP.B '2'                                            .00087800
674           OUTPUT = Y.STEP.C '3'                                            .00087900
675           OUTPUT = Y.STEP.D '4'                                            .00088000
    +         '  OR  ' Y.STEP.C '6'                                            .00088100
676           OUTPUT = Y.STEP.E '5' '  OR  ' Y.STEP.E '7'                      .00088200
677           OUTPUT = Y.STEP.F '6' '  OR  ' Y.STEP.F '4'                      .00088300
678           OUTPUT = Y.STEP.G '7  OR  ' Y.STEP.G '5'                         .00088400
679           OUTPUT =                                                         .00088500
680           L.STEP.A1 = IDENT(L.STEP.A1) P.U.                                .00088600
681           L.STEP.A2 = IDENT(L.STEP.A2) P.U.                                .00088700
682           L.STEP.B = IDENT(L.STEP.B) P.U.                                  .00088800
683           L.STEP.C.XA = IDENT(L.STEP.C.XA) P.U.                            .00088900
684           L.STEP.C.FA = IDENT(L.STEP.C.FA) P.U.                            .00089000
685           L.STEP.C.XB = IDENT(L.STEP.C.XB) P.U.                            .00089100
686           L.STEP.C.FB = IDENT(L.STEP.C.FB) P.U.                            .00089200
687           OUTPUT = 'THE ACTUAL LOOP PRECEDENCE IS'                         .00089300
688           OUTPUT = Y.STEP.A L.STEP.A1                                      .00089400
689           OUTPUT = Y.STEP.B L.STEP.A2                                      .00089500
690           OUTPUT = Y.STEP.C L.STEP.B                                       .00089600
691           OUTPUT = Y.STEP.D L.STEP.C.XA                                    .00089700
692           OUTPUT = Y.STEP.E L.STEP.C.FA                                    .00089800
693           OUTPUT = Y.STEP.F L.STEP.C.XB                                    .00089900
```

```
694             OUTPUT = Y.STEP.G L.STEP.C.FB                                 .00090000
695             OUTPUT = 'THE FINAL PLAUSIBILITY OF THIS MECHANIZATION IS:' .00090100
   +            '  ' CONVERT(PROB.SUCCESS,'INTEGER')     :(L0)                .00090150
696 N90         OUTPUT = 'FAILED TO FIND SELECTION PATTERN FOR NEW X & F'    .00090220
697             PROB.SUCCESS = PROB.SUCCESS * PROB.V     :(L90A)              .00090240
```

```
698 TIMEOUT       OUTPUT =                                        :(RETURN)          .
699 SCAN          POS = 1                                                           .00090500
700 SCAN.1        POS = IDENT(APAY<POS>) 0                        :S(FRETURN)       .00090600
701               IDENT(N.FCS<POS>)                               :S(SCAN.2)        .00090620
702               ARAY<PCS> PCS(0) ELEM RPOS(0)                   :F(SCAN.2)        .00090700
703               XPOS = N.PCS<POS>                                                 .00090720
704               N.POS<PCS> =                                    :(RETURN)         .00090740
705 SCAN.2        POS = POS + 1                                   :(SCAN.1)         .00090800
706 DECKOUT       W = 0                                                             .00090900
707               OUTPUT = 'THE DECK NOW IS:'                                       .00091000
708 DECKOUT.1     W = LT(W,I) W + 1                               :F(RETURN)        .00091100
709               OUTPUT = DUPL(' ',6 - SIZE(STN<W>)) STN<W> ' ' DECK<W>            .00091200
    +             DUPL(' ',60 - SIZE(DECK<W>))                                      .00091300
    +             DUPL(' ',20 - SIZE(LDECK<W>)) LDECK<W>                            .00091400
    +             DUPL(' ',25 - SIZE(UDECK<W>)) UDECK<W>                            .00091500
    +             DUPL(' ',05 - SIZE(W)) W                                          .00091700
    +             :(DECKOUT.1)                                                      .00091800
710 VAROUT        OUTPUT = 'XA = ' XA                                               .00091900
    +             DIFFEP(XA)                                                        .00091950
711               OUTPUT = 'THE INITIAL VALUE OF XA IS - ' XAR                      .00092000
    +             DIFFER(XAR)                                                       .00092050
712               OUTPUT = 'XP = ' XB                                               .00092100
    +             DIFFER(XB)                                                        .00092150
713               OUTPUT = 'THE INITIAL VALUE OF XB IS - ' XBR                      .00092200
    +             DIFFER(XBR)                                                       .00092250
714               OUTPUT = 'FA = ' FA                                               .00092300
    +             DIFFER(FA)                                                        .00092350
715               OUTPUT = 'FB = ' FB                                               .00092400
    +             DIFFER(FB)                                                        .00092450
716               OUTPUT = 'XN = ' XN                                               .00092500
    +             DIFFER(XN)                                                        .00092550
717               OUTPUT = 'XN IS CALCULATED FROM THE EXPRESSION:  '                .00092600
    +             DIFFER(FUNCTION.X) FUNCTION.X                                     .00092650
718               OUTPUT = 'FN = ' FN                                               .00092700
    +             DIFFER(FN)                                                        .00092750
719               OUTPUT = 'FUNCTION = ' FUNCTION                                   .00092800
    +             DIFFER(FUNCTION)                                                  .00092850
720               OUTPUT = 'THE FUNCTION REPRESINTS THE EXPRESSION:  '              .00092900
    +             DIFFER(FUNCTION.T) FUNCTION.T                   :(RETURN)         .00092950
721 VARIABLE      LE(SIZE(X),6)                                   :F(FRETURN)       .00093100
722               X POS(0) ANY(ALPHA) (SPAN(ALPHAMERIC) | NULL)                     .00093200
    +             RPOS(0)                   :F(FRETURN)S(RETURN)                    .00093300
723 REAL          X INT '.' (NULL | INT)                          :S(RETURN)        .00093400
    +                                                             F(FRETURN)        .00093500
724 SCAN.VAR      LL = 0                                                            .00093600
725 SCAN.VAR.1    LL  = LT(LL,10) LL + 1                          :F(FRETURN)       .00093700
726               IDENT(ST.PARM.F<LL>)                            :S(SCAN.VAR.1)    .00093800
727               IDENT(ST.PARM<LL>)                              :S(FRETURN)       .00093900
728               ST.PARM<LL> POS(0) XLL RPOS(0)                  :F(SCAN.VAR.1)    .00094000
729               ST.PARM.F<LL> =                                 :(RETURN)         .00094100
730 PROBOUT       OUTPUT = PS.ALL ' *' NAME '* '                                    .00094200
    +             PS.ALL.1 CONVERT(PROB.SUCCESS,'INTEGER') :(RETURN)                .00094300
731 SCAN.PARM     POS = 0                                                           .00094400
732               EQ(NPARM,0)                                     :S(FRETURN)       .00094500
733 SCAN.PARM.1   POS = LT(POS,NPARM) PCS + 1                     :F(FRETURN)       .00094600
734               PARM<PCS> PCS(C) PARM.X RPOS(0)                 :S(RETURN)        .00094700
```

```
    +                                                      F(SCAN.PARM.1)     .00094800
735 CONFLICT     IDENT(C.X)                               :F(CONFLICT.1)      .00094900
736              C.X = C.Y                                :(RETURN)           .00095000
737 CONFLICT.1   IDENT(C.X,C.Y)                           :S(RETURN)          .00095100
738              OUTPUT = 'THXRE EXISTS AN AMBIGUITY IN THE DEFINITION OF '   .00095200
    +            C.Z ' - - - WE ASSUME THE FIRST GUESS IS CORRECT'            .00095300
739              OUTPUT = DUPL(' ',5) 'THE PREVIOUS GUESS WAS - ' C.X         .00095400
740              OUTPUT = CUPL(' ',5) 'THE CURRENT  GUESS WAS - ' C.Y         .00095500
741              PROB.SUCCESS = PROB.SUCCESS * PROB.T     :(RETURN)           .00095600
742 FZERC        X (POS(0) ZERO RPOS(0))                 :S(RETURN)F(FRETURN) .00095700
743 REPLACE      TEMP2 =                                                      .00095800
744              TEMP1 = STRING                                               .00095900
745 REPLACE.1    TEMP1 (PCS(0) ARB NOTANY(ALPHAMERIC)) . TEMP3 IN             .00096000
    +            (BREAK('*/+-)') | RPCS(0)) =             :F(REPLACE.2)       .00096100
746              TEMP2 = TEMP2 TEMP3 OUT                  :(REPLACE.1)        .00096200
747 REPLACE.2    STRING = TEMP2 TEMP1                     :(RETURN)           .00096300
748 END                                                                       .00096400
```

```
COMPILATION TIME    3.717 SECONDS

MEMORY USAGE (DECIMAL BYTES)

CODE:               25046

STRINGS:             5533

VARIABLES:          13376

CONSTANTS:           3160

TOTAL:              47115

AVAILABLE:         120459

SUCCESSFUL COMPILATION
```

IBLIOGRAPHIC DATA HEET	1. Report No. UIUCDCS-R-73-562	2.	3. Recipient's Accession No.

. Title and Subtitle

ALGORITHM MODELS

5. Report Date
February 1973

6.

. Author(s)
DONALD JAY HOFFMAN

8. Performing Organization Rept. No. UIUCDCS-R-73-562

. Performing Organization Name and Address

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

10. Project/Task/Work Unit No.

11. Contract/Grant No.

2. Sponsoring Organization Name and Address

13. Type of Report & Period Covered
Thesis Research

14.

5. Supplementary Notes

16. Abstracts

The human mind performs in a way yet unknown. What I will try to do is treat the human mind as a black box and perform the same operations by creating a system of logical cognitive models of the required operations in taking an algorithm and translating it into a form suitable for coding. I will then demonstrate these cognitive models by demonstrating a method of checking the logical construction of implementations of some algorithms by using these models as part of a system for automatic fault detection.

17. Key Words and Document Analysis. 17a. Descriptors

algorithm
automatic fault detection
heuristic procedure

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement unlimited distribution	19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages 60
	20. Security Class (This Page UNCLASSIFIED	22. Price

OCT 29 1973

BOUND BY
HNM

UNIVERSITY OF ILLINOIS-URBANA
510.84 IL6R no. C002 no.559-564(1973
Modular microprogrammed computer with co
3 0112 088400566